N84-19814

Effects of Bearing Deadbands on Bearing Loads and Rotor Stability

Suite 1414, Executive Plaza, 555 Sparkman Drive, Huntsville, Alabama 35805, (205) 837-8510

J W. Col ED14

Effects of Bearing Deadbands on Bearing Loads and Rotor Stability

Final Report
on
Contract No. NAS8-35050

Prepared for:
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

Prepared by:

Control Dynamics Company
Suite 1414 Executive Plaza
555 Sparkman Drive
Huntsville, AL 35805
Telephone: (205) 837-8510

20 January 1984

ABSTRACT:

The stability properties and bearing loads observed in turbomachinery are significantly influenced by clearances (deadbands) in the load carriers which are usually ball bearings. In this study we have looked at a generic model of a turbopump, simplified to bring out these effects. This model demonstrates that bearing deadbands which are of the same order of magnitude or larger than the center-of-mass offset of a rotor due to mass imbalances cause significantly different dynamic behavior than would be expected of a linear, dynamical system. This fundamentally nonlinear behavior yields altered stability characteristics and altered bearing loading tendencies. It is shown that side forces can enhance system stability in the small, i.e. as long as the mass imbalance does not exceed some threshhold value or as long as no large, impulsive disturbances cause the motion to depart significantly from the region of stability. Limit cycles are investigated in this report and techniques for determining these limit cycles are developed. These limit cycles are the major source of bearing loading and appear in both synchronous and nonsynchronous forms. The synchronous limit cycles are driven by rotor imbalances. The nonsynchronous limit cycles (also called subsynchronous whirls) are self-excited and are the sources of instability. It is shown that such whirls are not necessarily unstable and can in fact be observed as relatively low level oscillations. They do, however, reveal the existence of an instability mechanism which may be waiting in the wings to destroy a machine if the speed is significantly increased or if the local stability conditions should cease to hold for some reason such as an impulsive disturbance. In this study we have shown that the nonlinear characteristics due to bearing deadbands have a significant effect on the dynamics of turbomachinery and cannot be ignored as has routinely been done in past analysis of such systems.

TABLE OF CONTENTS

1.0 INTRODUCTION

The problem of determining bearing loads and stability properties of rotating machines such as the turbopumps used in high performance rocket engines like the Space Shuttle Main Engine (SSME) is complex. Very high speeds are attained with significant fluid flows. As a consequence, bearing loads are potentially high with subsynchronous whirling likely. Typically, models used to analyze such systems are very complicated and nearly impossible to use for gaining insight into the basic phenomena involved. Linear models containing large numbers of degrees of freedom have been developed and applied to the analysis with mixed success. A significant nonlinearity is ignored by these models. The bearings typically have clearances of the order of .0005"-.0025". Since these machines are balanced to very high precision, the eccentricity of the rotor, i. e. the distance between the rotor center of mass and its geometric axis is of the same order or smaller in magnitude. Thus, bearing clearances or deadbands as they are more typically called, significantly affect the dynamics of these systems and must be taken into account. Taking this nonlinearity into account makes the analysis of the dynamics much more difficult. It is very desirable to have a simplified model of turbopump which retains the significant driving forces known to be present but readily lends itself to analysis. Such a model is available and is usually referred to as the Jeffcott model. We have modified this model by adding deadband effects along with fluid seal forces as currently understood. In addition, we have rewritten the equations of motion for the model in polar coordinates. This formulation is more naturally suited to the symmetry of the problem because the whirl orbits tend to be circular.

In addition to seal forces and deadbands, we have added a constant side force to the model to account for the likely misalignments between bearings and seals and also to account for hydrodynamic forces resulting from pumping fluids which may not be perfectly balanced due to slight imperfections in the internal geometry of the pump. The side force and deadband effects, working together, significantly affect the stability properties of the system in an interesting way. Stability may be enhanced under proper combinations but is only local stability in that it is possible to drive the system into instability by impulsive disturbances or large rotor imbalances.

The Jeffcott rotor is closer to reality than it may appear to the casual observer. Periodic synchronous or nonsynchronous orbiting motions of the rotor, referred to as whirls, are normally the motions of the system exhibited. Such an orbital motion can be described by a planar model. Thus, values for the effective mass, stiffness, deadband and seal coefficients can be found which will approximate the behavior of the more complex models. While exact frequencies of critical speeds and stability boundaries cannot be inferred from Jeffcott models, very good qualitative behavior can be investigated with these models and refined by higher fidelity hybrid simulations. For this reason, we consider the augmented Jeffcott model as the model of choice for developing an understanding of rotor whirl phenomena.

1.1 FORCE MODELS

The forces acting inside a turbopump are due to several causes. First, the fluid being pumped reacts upon the rotor with forces that are dependent upon rotor position and velocity and can be represented by linear models for small displacements [2]. The seals which prevent the high pressure fluid from leaking

away also generate forces on the rotor which can be modeled linearly. The assumed form representing these forces is given by

$$\underline{F}_{seal} = -C_s\underline{\dot{r}} - K_s\underline{r} + Q_s\,\underline{u}_X \times \underline{r} + C_Q\,\underline{u}_X \times \underline{\dot{r}} \tag{1.1}$$

These forces have the potential to drive whirl instability. Second, the force due to the mass eccentricity is a rotating force whose magnitude varies as the square of the rotor speed and is directed toward the rotor center of mass. This force is potentially destructive and must be minimized by stringent balancing of turbopump rotors. The form of this force is as shown below:

$$\underline{F}_e = -m\,\omega^2\,\underline{\varepsilon} \tag{1.2}$$

Turbopump rotors are maintained in position by bearing forces. These bearing forces are generated by a rather complex interaction involving bending forces of the rotor shaft, the deformation of the bearing balls or rollers, the motion and deformation of the bearing races, the bearing retainers, the bearing carriers etc. Detailed modeling of these interactions is the subject of numerous complex analyses. For our purposes, we shall assume that the bearing itself tends to act as a linear spring. However, clearances between bearing races and carriers or shafts allow some small region of free motion of the rotor shaft relative to its housing. For simplicity, we idealize the bearing balls or rollers as a uniform annular ring separating the rotor shaft and housing. The effective surface roughness of the contacting surfaces provides some initial low stiffness values for the bearing system. As the surfaces come closer together the apparent stiffness increases resulting in the force curve shown in Figure 1.1. This bearing force curve is further idealized for our analyses and can be expressed by

$$\underline{F}_B = \begin{cases} K_B\,(\underline{r} - g\,\underline{u}_r) & |\underline{r}| > g \\ 0 & |\underline{r}| \leq g \end{cases} \tag{1.3}$$

These forces represent the set of forces believed to be significant to the determination of the dynamic characteristics of turbopump rotors. We shall now proceed to consider the dynamic behavior of a system driven by these forces.

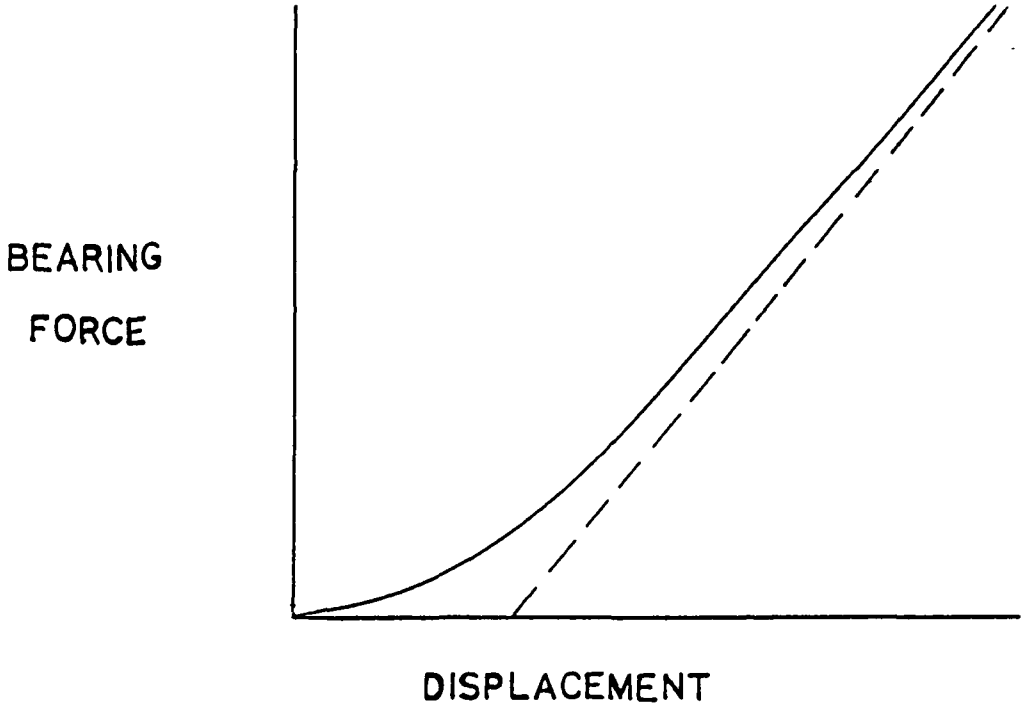

Figure 1.1 Bearing Force Curve.

2.0 DESCRIPTION OF ANALYSIS METHODS

2.1 MODEL DESCRIPTION AND DERIVATION OF THE EQUATIONS OF MOTION

To investigate the effects of bearing deadbands on system stability and on bearing loads, the Jeffcott rotor model [2] was chosen. Several additions to this simple model have been made to include the effects of mechanisms present in a turbopump that are not reflected in the basic model. The effects of a deadband in the system are considered as well as those of a side force, and seal forces.

Shown in Figure 2.1 is a diagram of the assumed geometry of the model used. The vector $\underline{r}$ is the displacement of the rotor center from its equilibrium position (rotor at rest). The angle ϕ is the angle made by $\underline{r}$ with the y axis, (the whirl angle). The imbalance in the rotor is represented by the vector $\underline{\varepsilon}$, the magnitude of which is constant. $\underline{\varepsilon}$ is the shaft eccentricity, the displacement of the shaft center of mass from the grometric center. The angular velocity of the shaft is represented by ω. This shaft speed is considered to be constant in the analysis. The quantity g is the deadband present between the shaft and the bearing. It is not shown in the figure.

Forces which must be considered in formulating the equations of motion for the system include bearing forces, seal forces, imbalance forces and side forces. The vector force diagram in Figure 2.2 indicates these forces and the directions in which they act. The relative magnitudes of the vectors as drawn in the diagram are not precise. The forces present in the system due to the seal arise from the seal stiffness, K_S, the seal damping, C_S, the cross coupling stiffness, Q_S, and the cross coupling damping, C_Q. The bearing forces result from the bearing stiffness, K_B. Note the if the magnitude of $\underline{r}$ is less than the deadband, then the bearing forces will be zero.

The equations of motion for the system are derived in both polar and cartesian coordinates. The unit vectors $\underline{u}_r$, $\underline{u}_\phi$, and $\underline{u}_x$ in Figure 2.1 indicate the reference frame for the polar coordinate derivation. The cartesian reference frame is indicated by the x, y, and z axes. Given below are the equations of motion in polar coordinates followed by the system description in cartesian coordinates. The only nonlinearity in the cartesian coordinate description is due to the presence of the deadband.

The force in the system due to the bearing reaction with the rotor may be expressed as

$$\underline{F}_B = \begin{cases} K_B\,(\underline{r} - g\underline{u}_r) & |\underline{r}| > g \\ 0 & |\underline{r}| \leq g \end{cases} \qquad (2.1)$$

with $\underline{r} = r\underline{u}_r$.

Forces due to the seals are expressed as

$$\underline{F}_{seal} = -C_S\,\dot{\underline{r}} - K_S\,\underline{r} + Q_S\,\underline{u}_x \times \underline{r} + C_Q\,\underline{u}_x \times \dot{\underline{r}} \qquad (2.2)$$

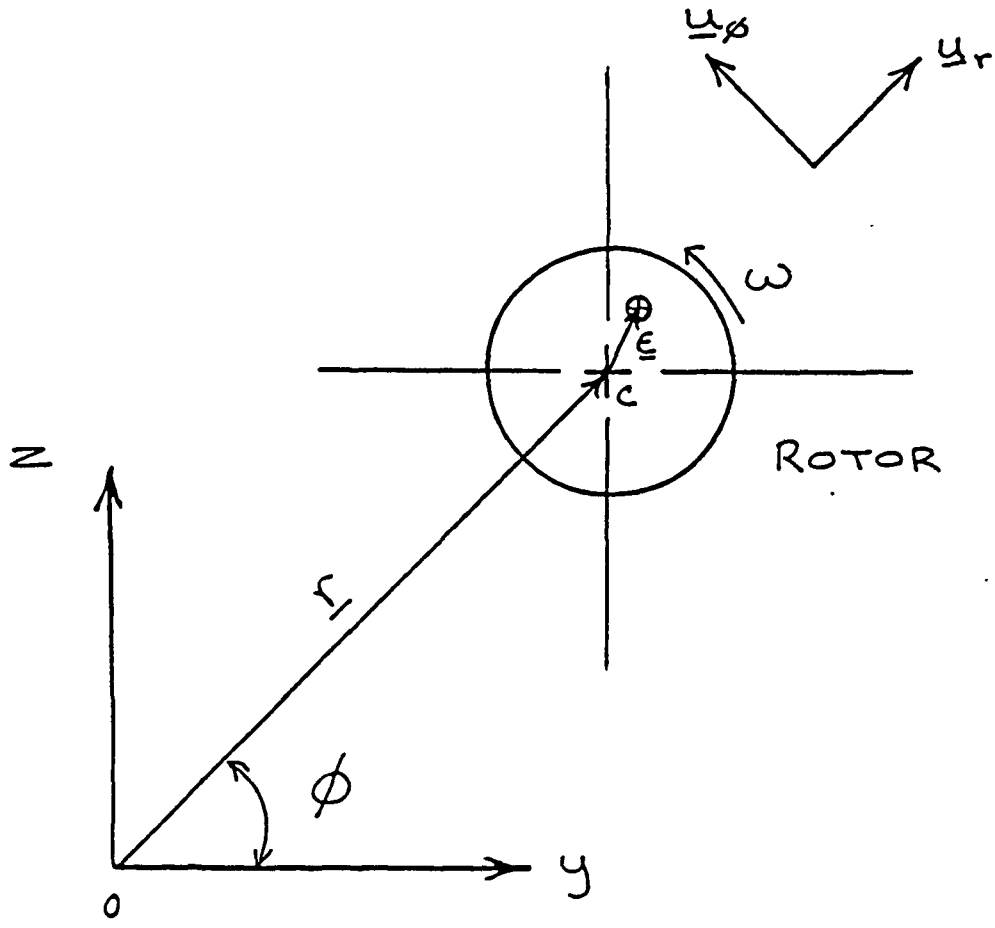

Figure 2.1 Assumed Geometry for Deriving the Equations of Motion.

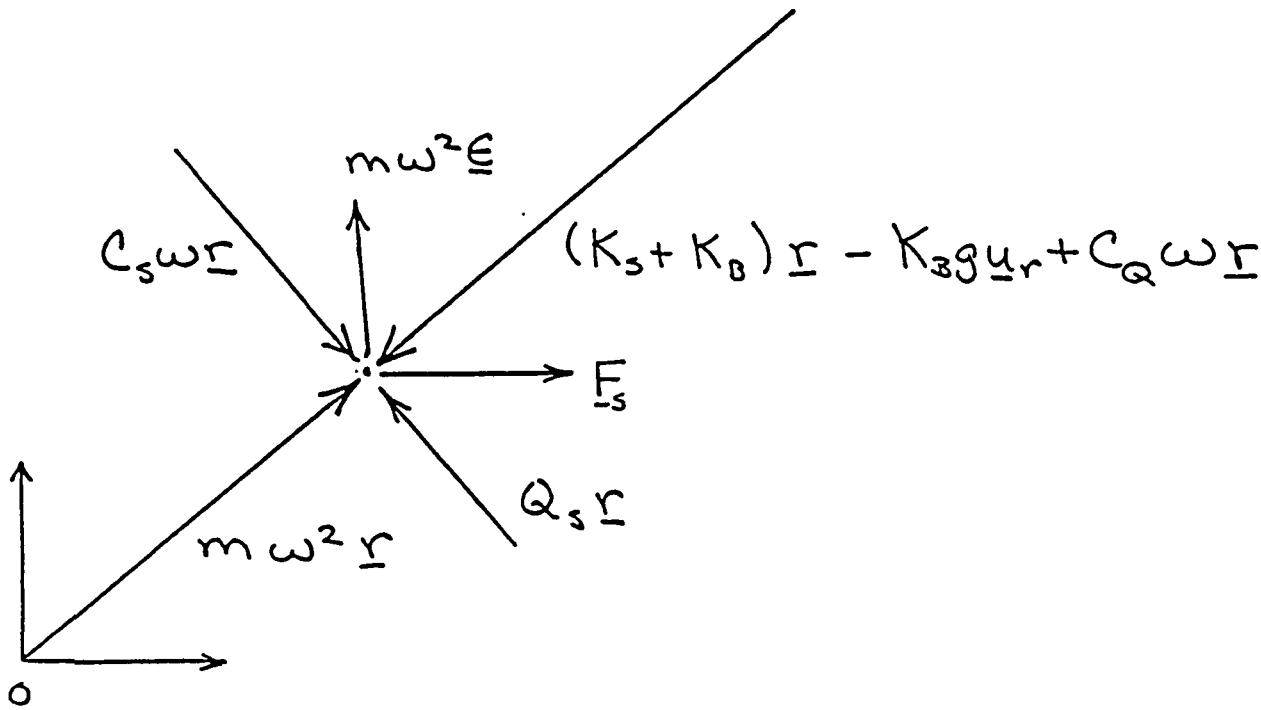

Figure 2.2 Vector Force Diagram - Not to Scale.

where $\underline{u}_x$ is the unit vector oriented axially along the shaft. The side force is expressed as the variable $\underline{F}_S$. The force equation for the system is:

$$m(\ddot{\underline{r}} + \ddot{\underline{\varepsilon}}) = \sum_i \underline{F}_i \quad , \tag{2.3}$$

with $\sum F_i$ representing the forces due to seals, bearings, and side forces. The imbalance acceleration $\ddot{\underline{\varepsilon}}$ expressed as follows:

$$\dot{\underline{\varepsilon}} = \underline{\omega} \times \underline{\varepsilon}$$

$$\ddot{\underline{\varepsilon}} = \underline{\omega} \times (\underline{\omega} \times \underline{\varepsilon}) \tag{2.4}$$

or

$$\ddot{\underline{\varepsilon}} = -\omega^2 \underline{\varepsilon}$$

Equation 2.3 may now be written as

$$m\ddot{\underline{r}} = \sum_i \underline{F}_i + m\omega^2 \underline{\varepsilon} \tag{2.5}$$

Insertion of the expressions in Equations 2.2 and 2.3 and inclusion of the side force term yields:

$$m\ddot{\underline{r}} = -K_B (\underline{r} - g\underline{u}_r) - K_S \underline{r} - C_S \dot{\underline{r}} + Q_S \underline{u}_x \times \underline{r} + C_Q \underline{u}_x \times \dot{\underline{r}} + \underline{F}_S + m\omega^2 \underline{\varepsilon}. \tag{2.6}$$

The vector derivatives and cross products are:

$$\dot{\underline{u}}_r = \dot{\phi}\, \underline{u}_\phi \tag{2.7}$$

$$\dot{\underline{u}}_\phi = -\dot{\phi}\, \underline{u}_r \tag{2.8}$$

$$\underline{u}_x \times \underline{u}_r = \underline{u}_\phi \tag{2.9}$$

$$\dot{\underline{r}} = \dot{r}\underline{u}_r + r\dot{\phi}\underline{u}_\phi \tag{2.10}$$

$$\ddot{\underline{r}} = (\ddot{r} - r\dot{\phi}^2)\underline{u}_r + (2\dot{r}\dot{\phi} + r\ddot{\phi})\underline{u}_\phi \tag{2.11}$$

Equation 2.6 may, therefore, be expressed as:

$$m \begin{bmatrix} \ddot{r} - r\dot{\phi}^2 \\ 2\dot{r}\dot{\phi} + r\ddot{\phi} \end{bmatrix} = -K_B \begin{bmatrix} r - g \\ 0 \end{bmatrix} - K_S \begin{bmatrix} r \\ 0 \end{bmatrix} - C_S \begin{bmatrix} \dot{r} \\ r\dot{\phi} \end{bmatrix} + Q_S \begin{bmatrix} 0 \\ r \end{bmatrix}$$

$$+ C_Q \begin{bmatrix} -r\dot{\phi} \\ \dot{r} \end{bmatrix} + F_S \begin{bmatrix} \cos\phi \\ -\sin\phi \end{bmatrix} + m\omega^2 \begin{bmatrix} \varepsilon_r \\ \varepsilon_\phi \end{bmatrix} \quad , \tag{2.12}$$

(with the unit vectors $\underline{u}_r$ and $\underline{u}_\phi$ implied), from which the following two differential equations are obtained:

$$\ddot{r} = \frac{-K_B}{m}(r-g) - \frac{K_S}{m} r - \frac{C_S}{m} \dot{r} - \frac{C_Q}{m} r\dot{\phi} + \frac{F_S}{m} \cos\phi + r\dot{\phi}^2 + \omega^2 \varepsilon \cos(\omega t - \phi) \tag{2.13}$$

$$\ddot{\phi} = \frac{Q_S}{m} - \frac{C_S}{m}\dot{\phi} + \frac{C_Q}{m}\frac{\dot{r}}{r} - \frac{F_S}{m}\sin\phi - \frac{2\dot{r}\dot{\phi}}{r} + \frac{\omega^2\varepsilon}{r}\sin(\omega t - \phi) \tag{2.14}$$

The equations of motion in cartesian coordinates may be derived in a similar manner. The bearing forces are:

$$F_{By} = \frac{-K_B(|\underline{r}| - g)}{|\underline{r}|} y \quad \text{for } |\underline{r}| > g \tag{2.15}$$

$$F_{Bz} = \frac{-K_B(|\underline{r}| - g)}{|\underline{r}|} z \quad \text{for } |\underline{r}| > g \tag{2.16}$$

where

$$|\underline{r}| = \sqrt{y^2 + z^2}\ . \tag{2.17}$$

Again, if $|\underline{r}| < g$, then $F_B = 0$. The seal forces are expressed as:

$$F_{Sy} = -K_S y - C_S\dot{y} - Q_S z - C_Q\dot{z} \tag{2.18}$$

$$F_{Sz} = -K_S z - C_S\dot{z} + Q_S y + C_Q\dot{y} \tag{2.19}$$

The force equations are:

$$m(\ddot{y} + \ddot{\varepsilon}) = \sum_i F_{yi} + F_S \quad \text{and} \quad m(\ddot{z} + \ddot{\varepsilon}) + \sum_i F_{zi}. \tag{2.20}$$

The y and z components of the imbalance term are:

$$\varepsilon_y = -\omega^2\varepsilon\cos\omega t \quad \text{and} \quad \varepsilon_z = -\omega^2\varepsilon\sin\omega t \tag{2.21}$$

The two differential equations describing the system may now be formed.

$$\ddot{y} = \frac{K_B(|\underline{r}| - g)}{m|\underline{r}|} y - \frac{K_S}{m} y - \frac{C_S}{m}\dot{y} - \frac{Q_S}{m} z - \frac{C_Q}{m}\dot{z} + \frac{F_S}{m} + \omega^2\varepsilon\cos\omega t \tag{2.22}$$

$$\ddot{z} = \frac{-K_B(|\underline{r}| - g)}{m|\underline{r}|} z - \frac{K_S}{m} z - \frac{C_S}{m}\dot{z} + \frac{Q_S}{m} y + \frac{C_Q}{m}\dot{y} + \omega^2\varepsilon\sin\omega t \tag{2.23}$$

Because the side forces are assumed to act only in the y direction, no side force term appears in the $\ddot{z}$ equation.

2.2 SIMULATION OF THE SIMPLE ROTOR SYSTEM

Having derived the equations of motion for the system, solutions for the states may be obtained by solving the resulting differential equations. This is accomplished by first casting the two second order differential equations into a form easily solved by numerical integration. By making the following definitions for r, $\dot{r}$, ϕ, and $\dot{\phi}$, the system may be written in the form of four first order differential equations.

Let

$$p_1 = r \qquad p_3 = \phi$$
$$p_2 = \dot{r} \qquad p_4 = \dot{\phi} \tag{2.24}$$

Then

$$\dot{p}_1 = p_2 \tag{2.25}$$

$$\dot{p}_2 = \frac{-(K_B + K_S)}{m} p_1 + K_B g - \frac{C_S}{m} p_2 - \frac{C_Q}{m} p_1 p_4 + \frac{F_S}{m} \cos p_3 \tag{2.26}$$
$$+ \omega^2 \varepsilon \cos (\omega t - p_3) + p_1 + p_4^2$$

$$\dot{p}_3 = p_4 \tag{2.27}$$

$$\dot{p}_4 = \frac{Q_S}{m} - \frac{C_S}{m} p_4 + \frac{C_Q p_2}{m p_1} - \frac{F_S}{m p_1} \sin p_3 + \frac{\omega^2 \varepsilon}{p_1} \sin (\omega t - p_3) - \frac{2 p_2 p_4}{p_1} \tag{2.28}$$

A similar procedure may be followed to express the cartesian coordinate equations of motion in state variable form.

Let

$$q_1 = y \qquad q_3 = z$$
$$q_2 = \dot{y} \qquad q_4 = \dot{z} \tag{2.29}$$

and $|\underline{r}| = \sqrt{q_1^2 + q_3^2}$

Then

$$\dot{q}_1 = q_2 \tag{2.30}$$

$$\dot{q}_2 = \frac{-K_B (|\underline{r}| - g)}{m |\underline{r}|} q_1 - \frac{K_S}{m} q_1 - \frac{C_S}{m} q_2 - \frac{Q_S}{m} q_3 - \frac{C_Q}{m} q_4 + \frac{F_S}{m} + \omega^2 \varepsilon \cos \omega t \tag{2.31}$$

$$\dot{q}_3 = q_4 \tag{2.32}$$

$$\dot{q}_4 = \frac{-K_B (|\underline{r}| - g)}{m |\underline{r}|} q_3 - \frac{K_S}{m} q_3 - \frac{C_S}{m} q_4 + \frac{Q_S}{m} q_1 + \frac{C_Q}{m} q_2 + \omega^2 \varepsilon \sin \omega t \tag{2.33}$$

The system is now amenable to solution by numerical methods. The method chosen to solve for the various states of the system is a Runge-Kutta iterative method. A simulation based on this method has been compiled and used extensively to examine the properties of the deadband rotor model. Basically, the simulation is

composed of a driver program for the integrations, and a subroutine containing the four first order differential equations. Simulations for each of the coordinate systems considered have been generated and compared for equivalent cases with good agreement. Because the system lends itself so well to analysis in polar coordinates, this version of the simulation is used most extensively.

To examine the character of the resulting whirl orbits we performed numerous parameter studies using time and frequency plots to evaluate the behavior. The frequency spectrum of the orbit indicates whether it is a synchronous or subsynchronous whirl.

Three different orbital types have been defined for a rotor system. These have been given the names A-type, B-type, and C-type motion. Each may be characterized by the shape of the orbit, its orientation with respect to the system origin, and its frequency.[3]

For A-type motion, the rotor has moved to an "equilibrium" position, other than the rotor rest position, and the orbit is about this point. These orbits about the equilibrium point are typically small in radius compared to the deadband; however, they may be rather large with respect to the deadband magnitude depending on the shaft spin rate and the imbalance present in the system. An A-type orbit does not encircle the origin of the coordinate system (the rotor rest position). The whirl orbit may be totally inside the deadband area, partially within the deadband area, or totally outside the deadband, as indicated by the sample plots of A-type whirl orbits shown in Figures 2.3 through 2.8. The run identification number appearing on each plot will be explained shortly. These numbers indicate the system parameter values used in a particular run. The solid curve in the plot is representative of the motion of the center of the rotor shaft with the dashed line representing the deadband. This convention is employed in all plots of shaft orbits presented in this report. The identification code used to label each plot will be explained shortly. A-type whirl is typically at synchronous speed as indicated by the acompanying PSD plots shown with each orbit plot. The power spectral density (PSD) plots have been rescaled because only relative magnitudes are important. In the PSD plots there are three curves. This is because the PSD of the radius vector as well as its two components has been performed. A-type whirls occur in a variety of shapes and sizes.

B-type motion is rather random with the rotor bouncing around inside the deadband area. This motion may resemble A-type or C-type, however the motion is not strictly periodic. The major frequency component of this type of orbit is generally nonsynchronous, with synchronous frequencies and multiples thereof also present. Figures 2.9 through 2.11 indicate several B-type orbits shown with their corresponding PSD plots. The orbits shown in Figures 2.3 and 2.12 have the same run identification number but appear different from each other because the runs were executed with different shaft speeds.

C-type motion surrounds all or most of the deadband area and always encircles the origin of the coordinate system. This type of orbit may be at synchronous or subsynchronous frequency, depending upon various system parameters. In most cases considered in this study the frequency of the C-type orbits are half-synchronous as are the cases presented in Figures 2.12 through 2.14. The PSD plots for these C-type orbits accompany their plots.

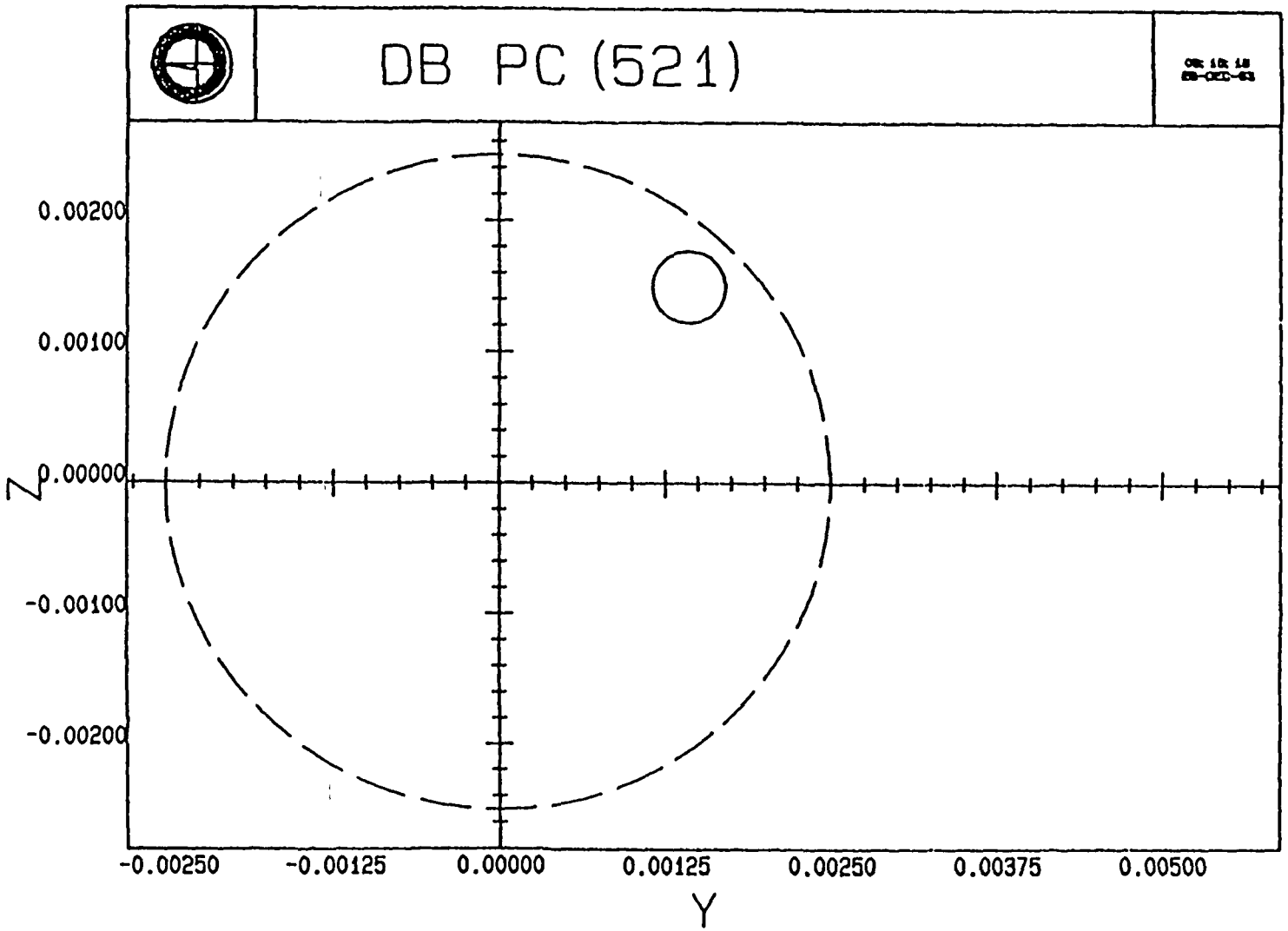

Figure 2.3a A-Type Motion Within Deadband.

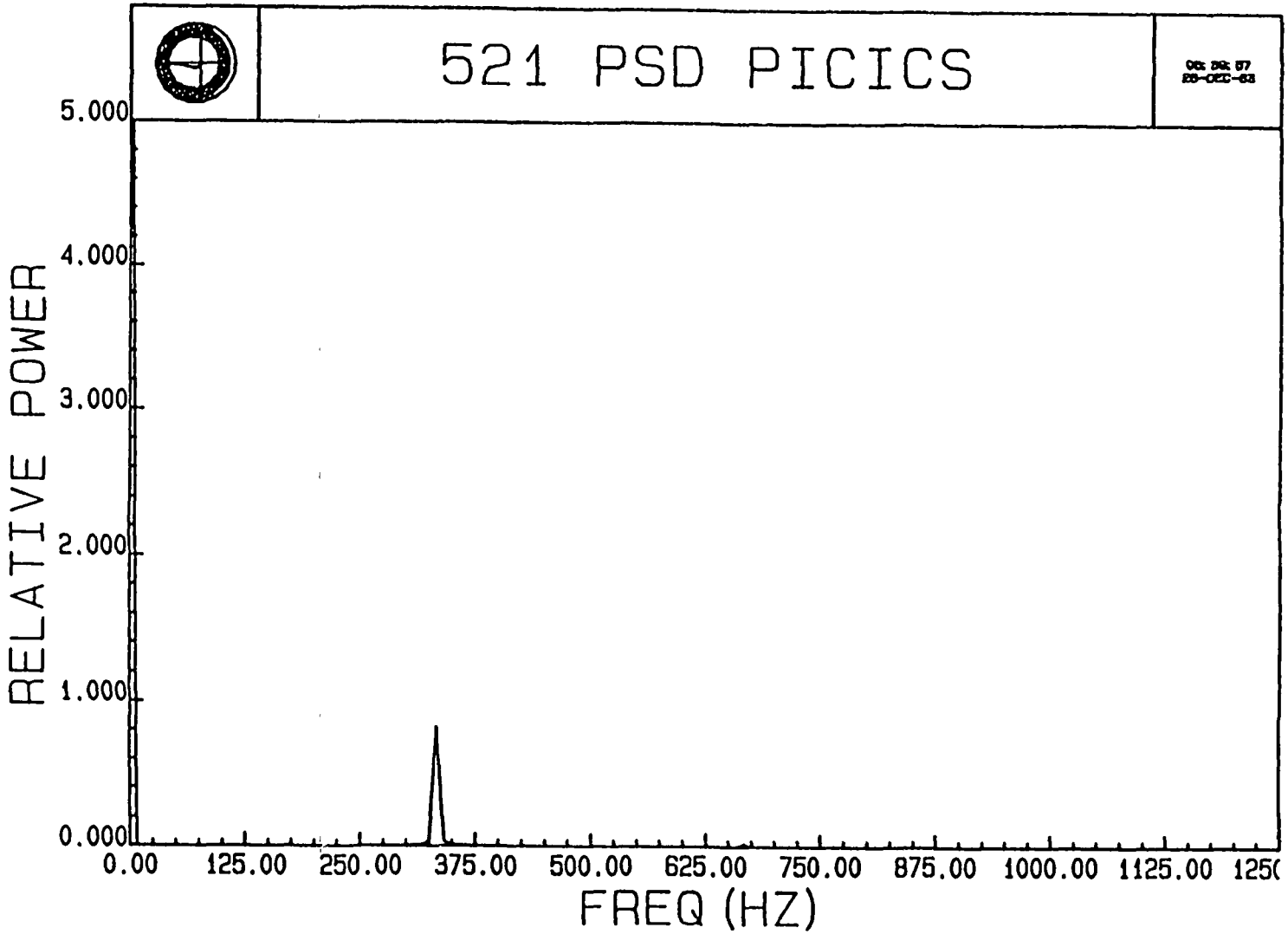

Figure 2.3b PSD for 2.3a; Shaft Speed: 333 Hz.

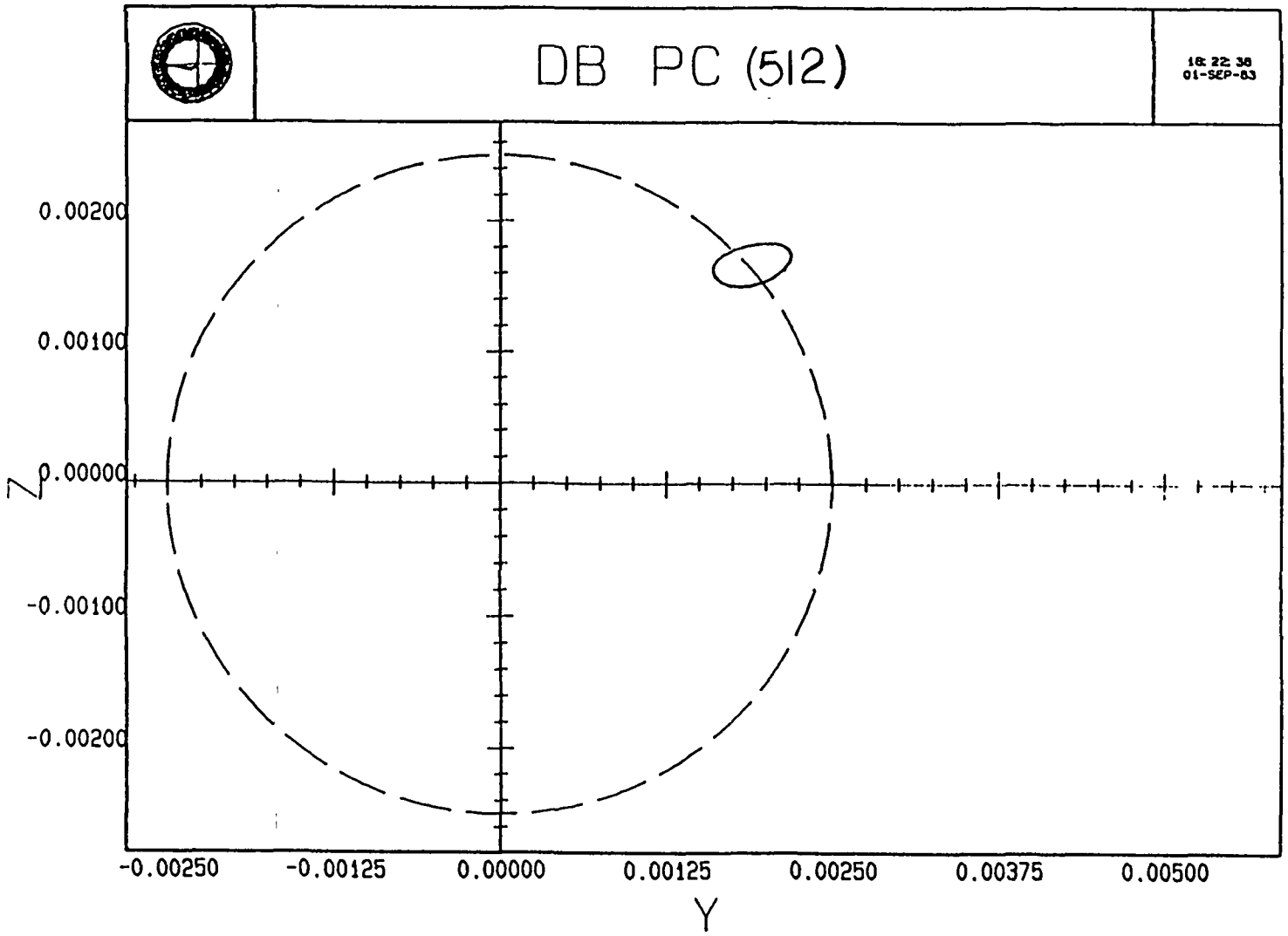

Figure 2.4a A-Type Motion Overlapping Deadband.

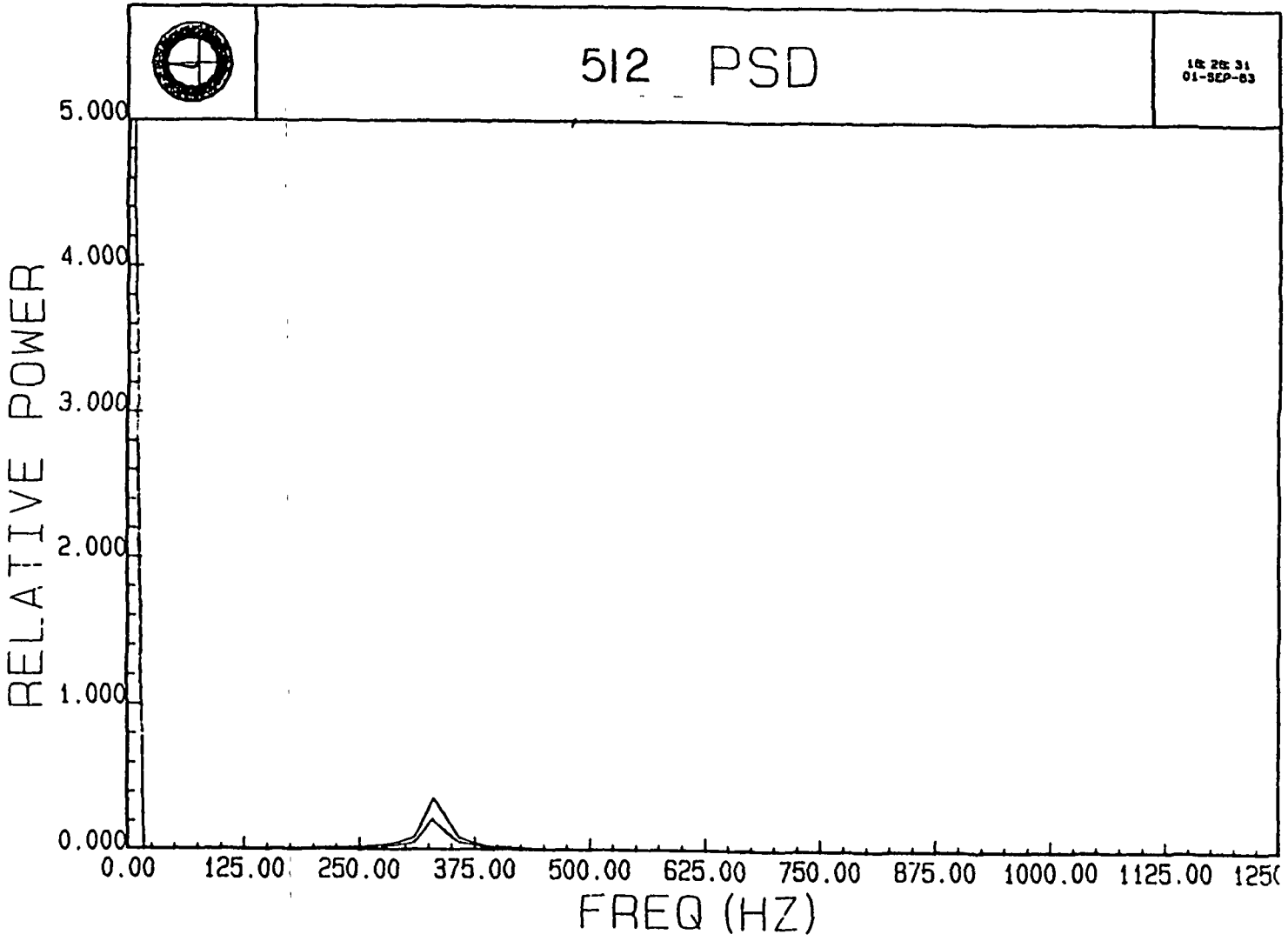

Figure 2.4b PSD for 2.4a; Shaft Speed: 333 Hz.

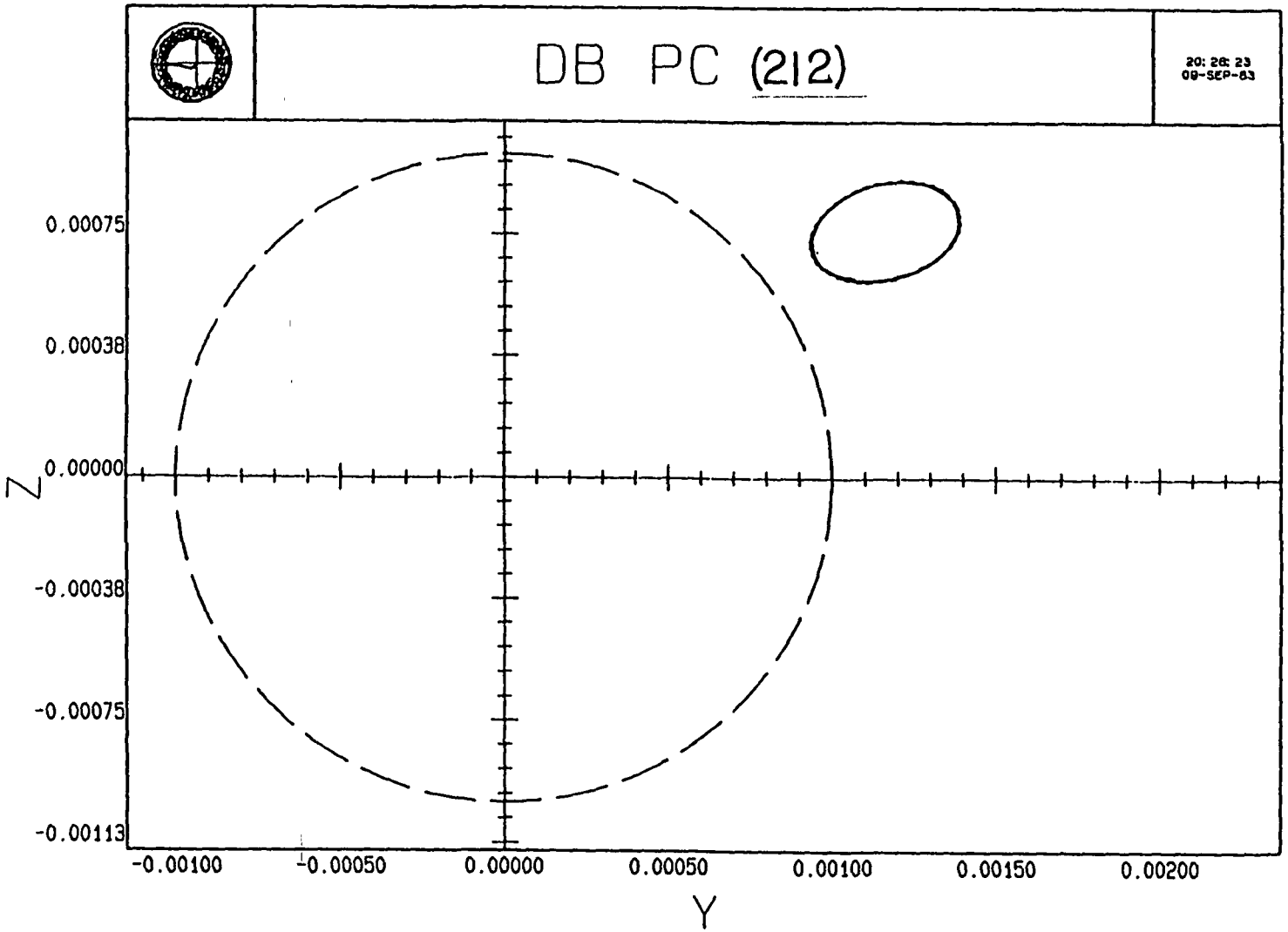

Figure 2.5a A-Type Motion Outside Deadband.

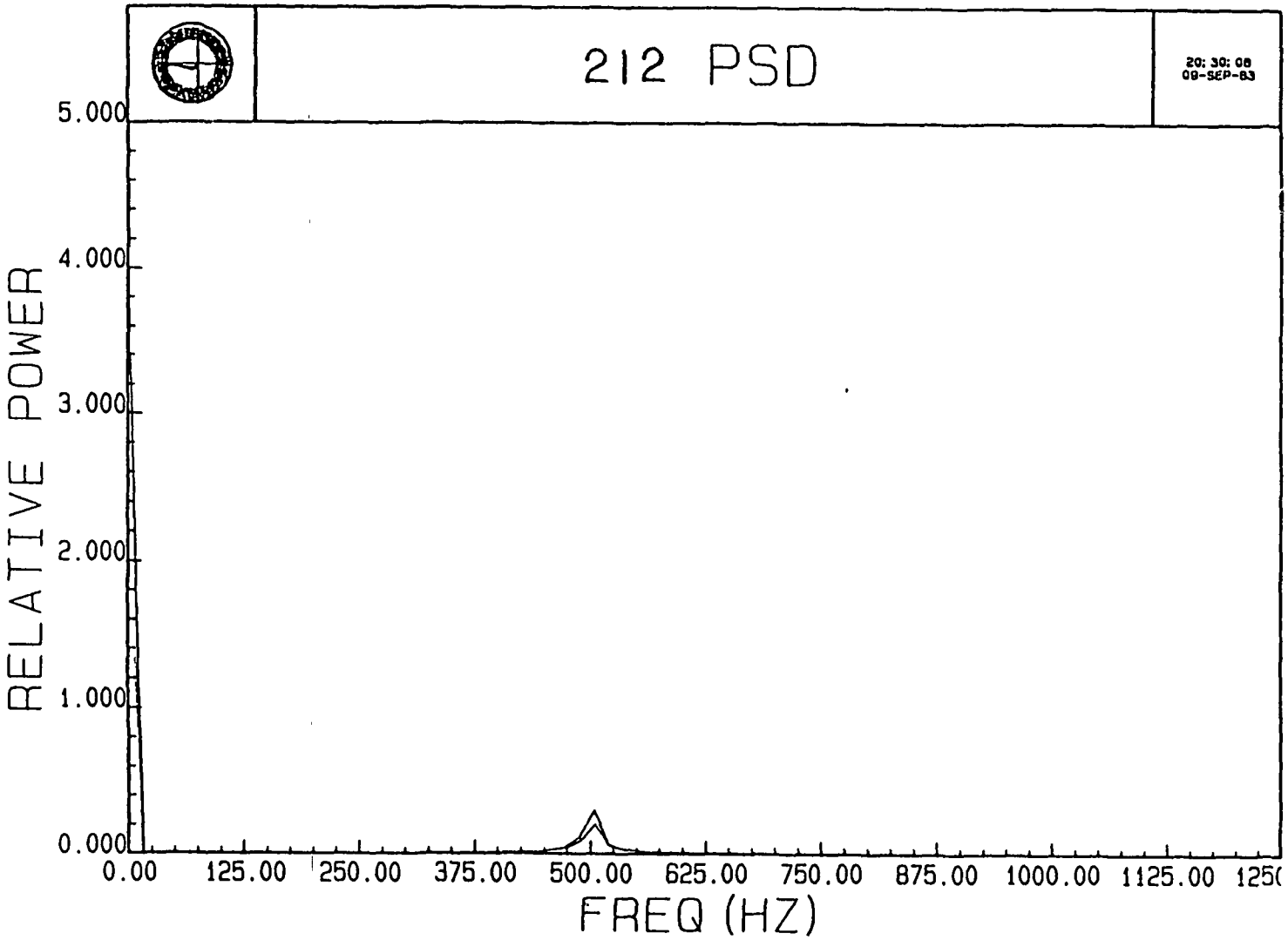

Figure 2.5b PSD for 2.5a; Shaft Speed: 500 Hz.

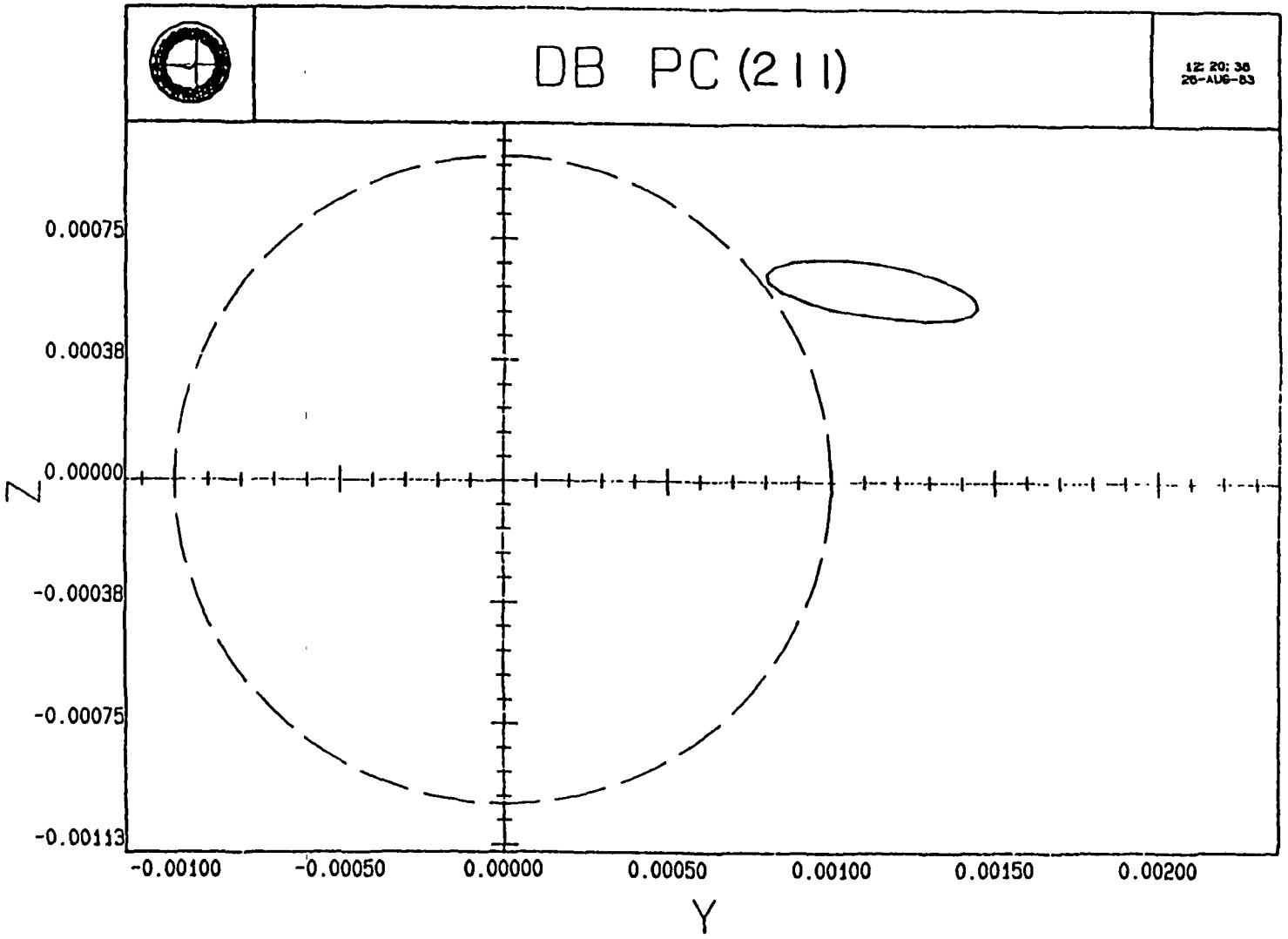

Figure 2.6a A-Type Motion Outside Deadband.

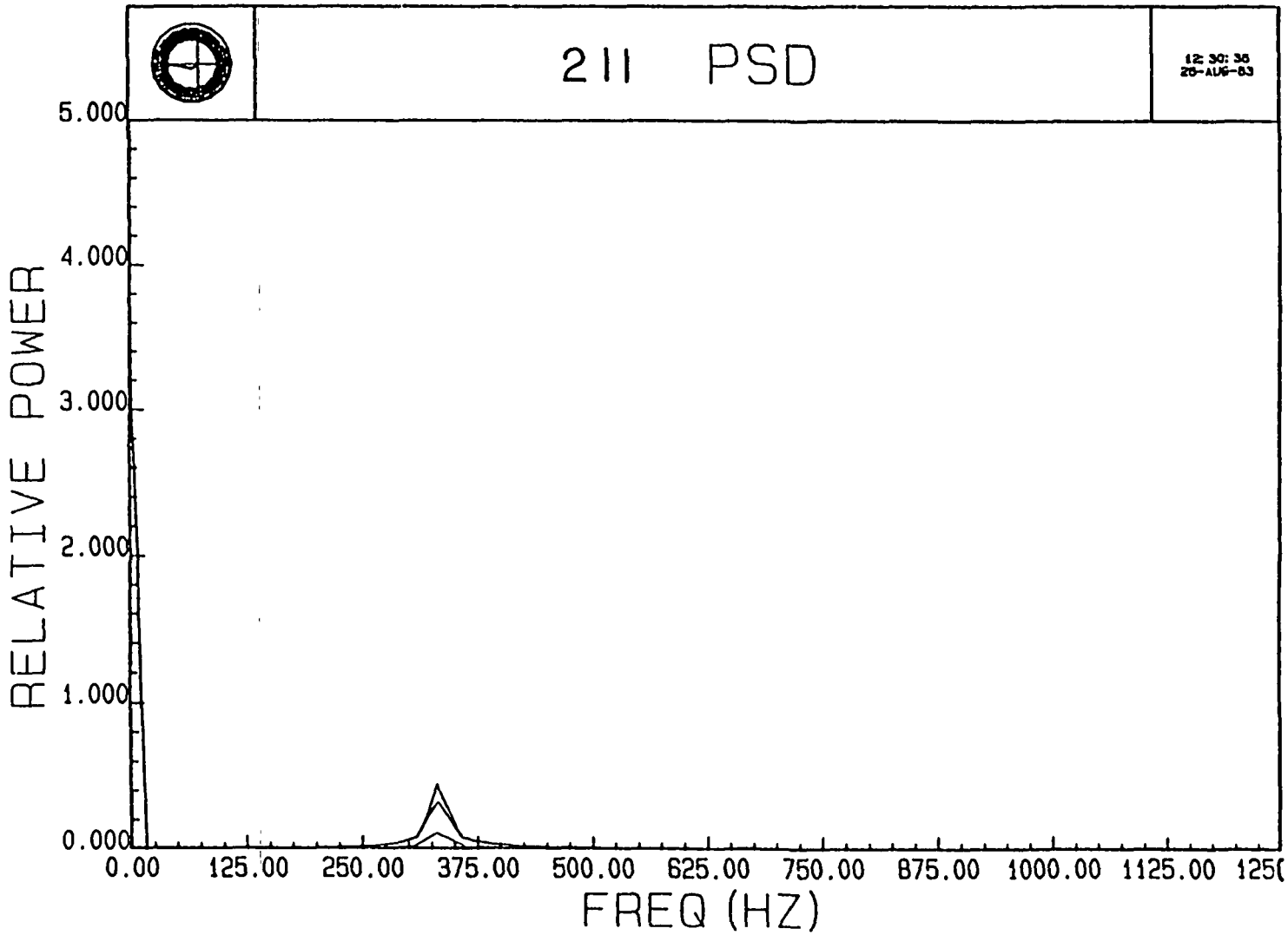

Figure 2.6b PSD for 2.6a; Shaft Speed: 333 Hz.

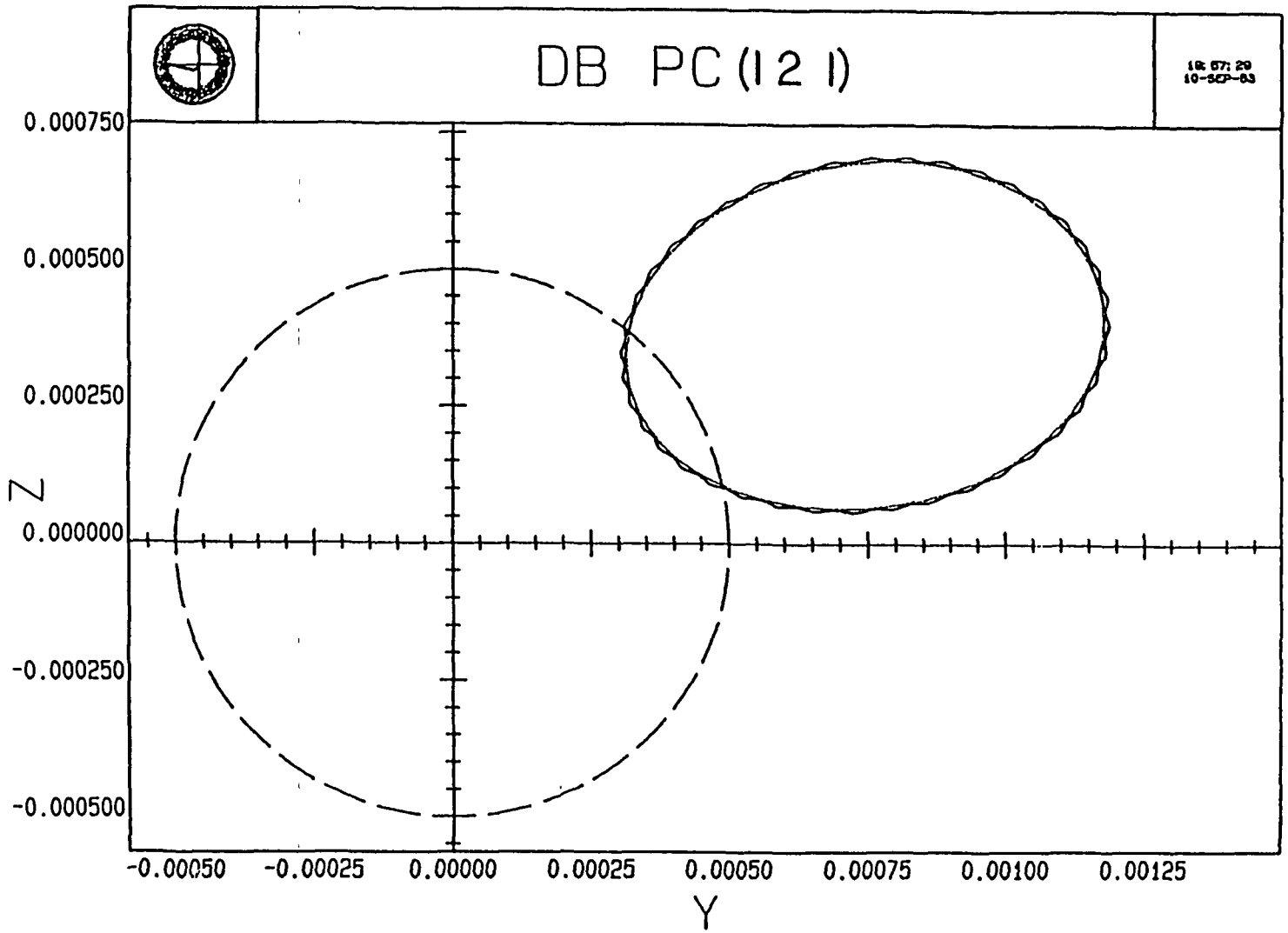

Figure 2.7a A-Type Motion, A Rather Large Orbit Overlapping the Deadband.

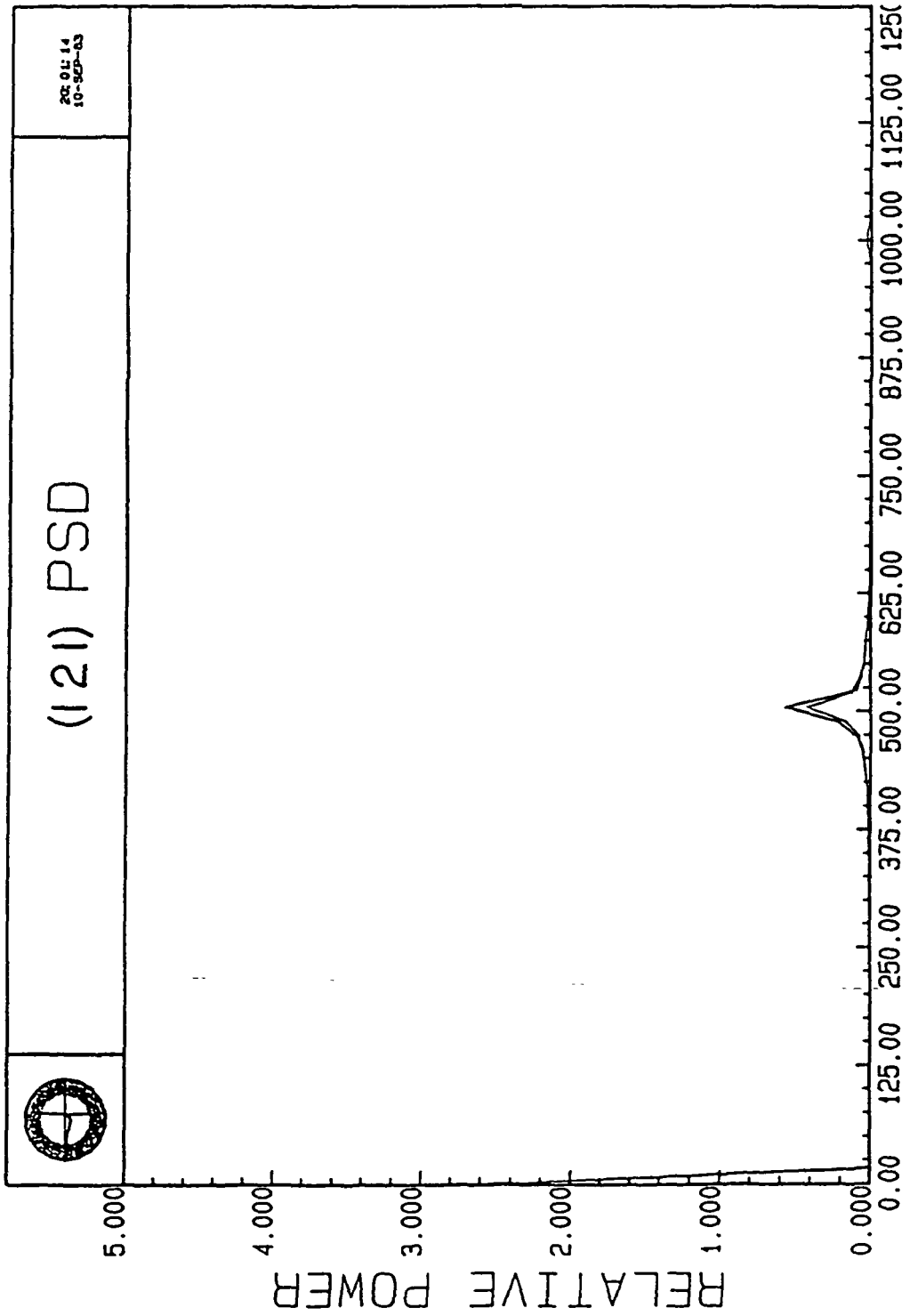

Figure 2.7b PSD for 2.7a; Shaft Speed: 500 Hz.

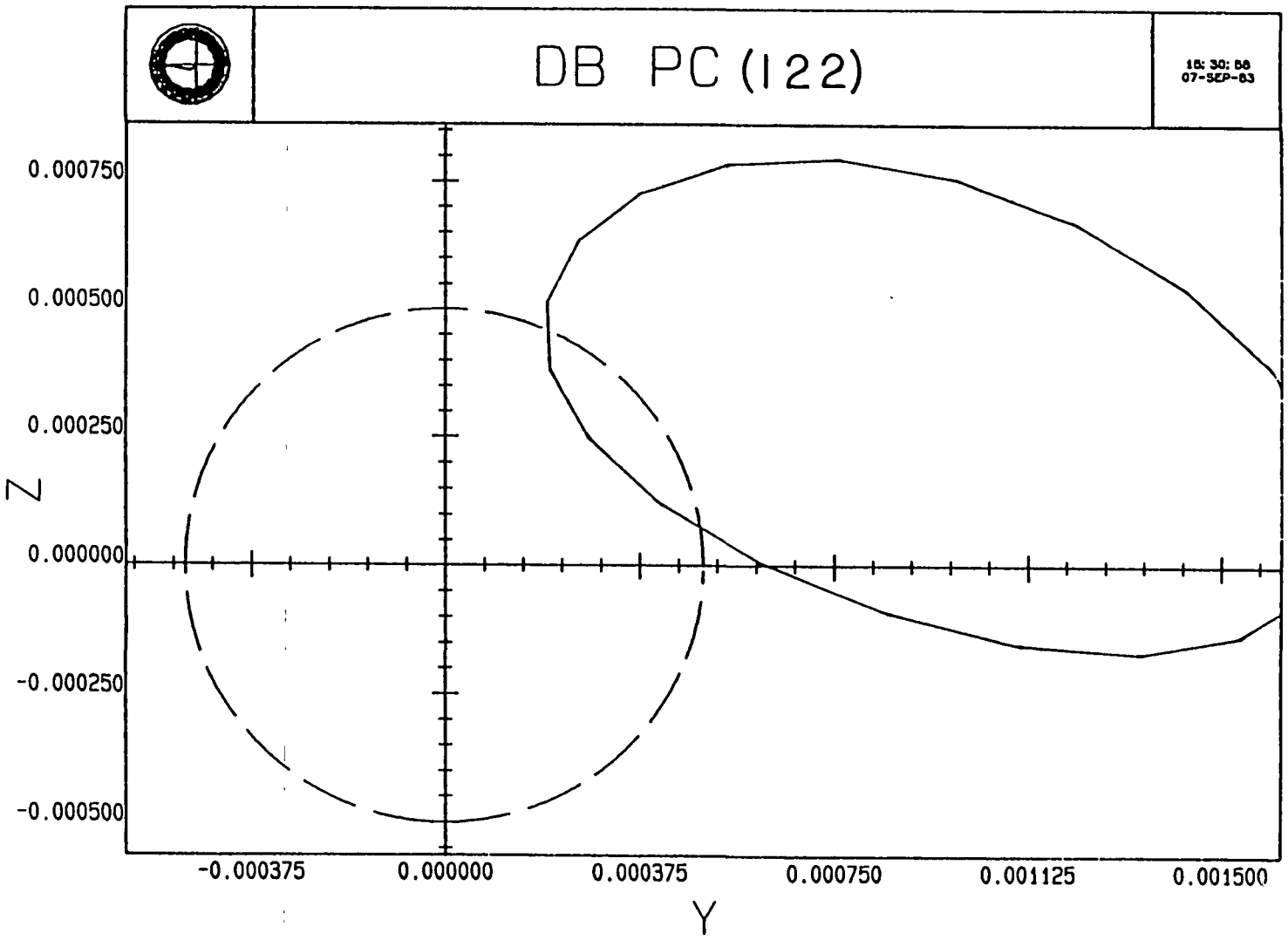

Figure 2.8a A-Type Motion; A Very Large Orbit Overlapping the Deadband.

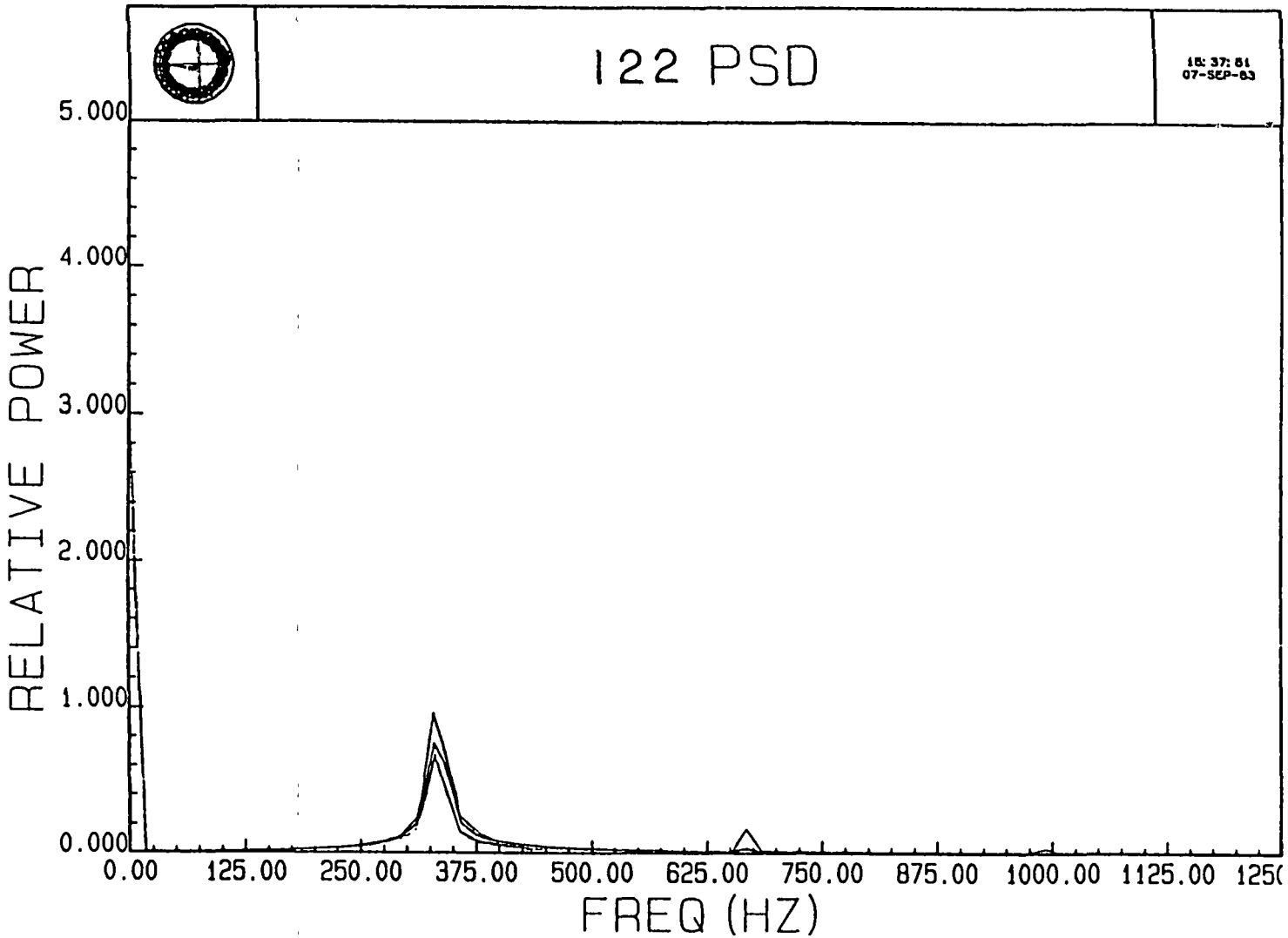

Figure 2.8b PSD for 2.8a; Shaft Speed: 333 Hz.

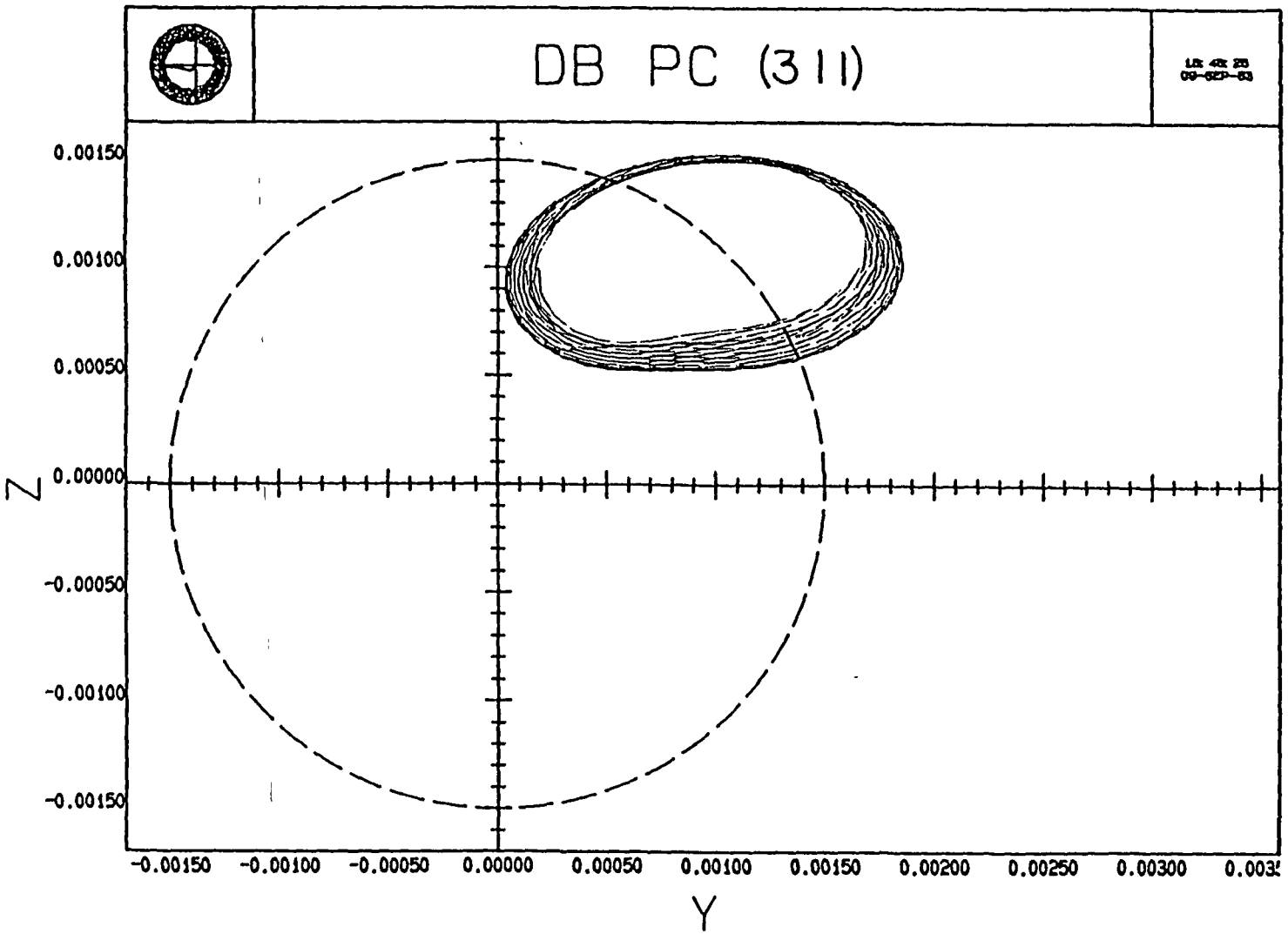

Figure 2.9a B-Type Motion Confined to First Quandrant.

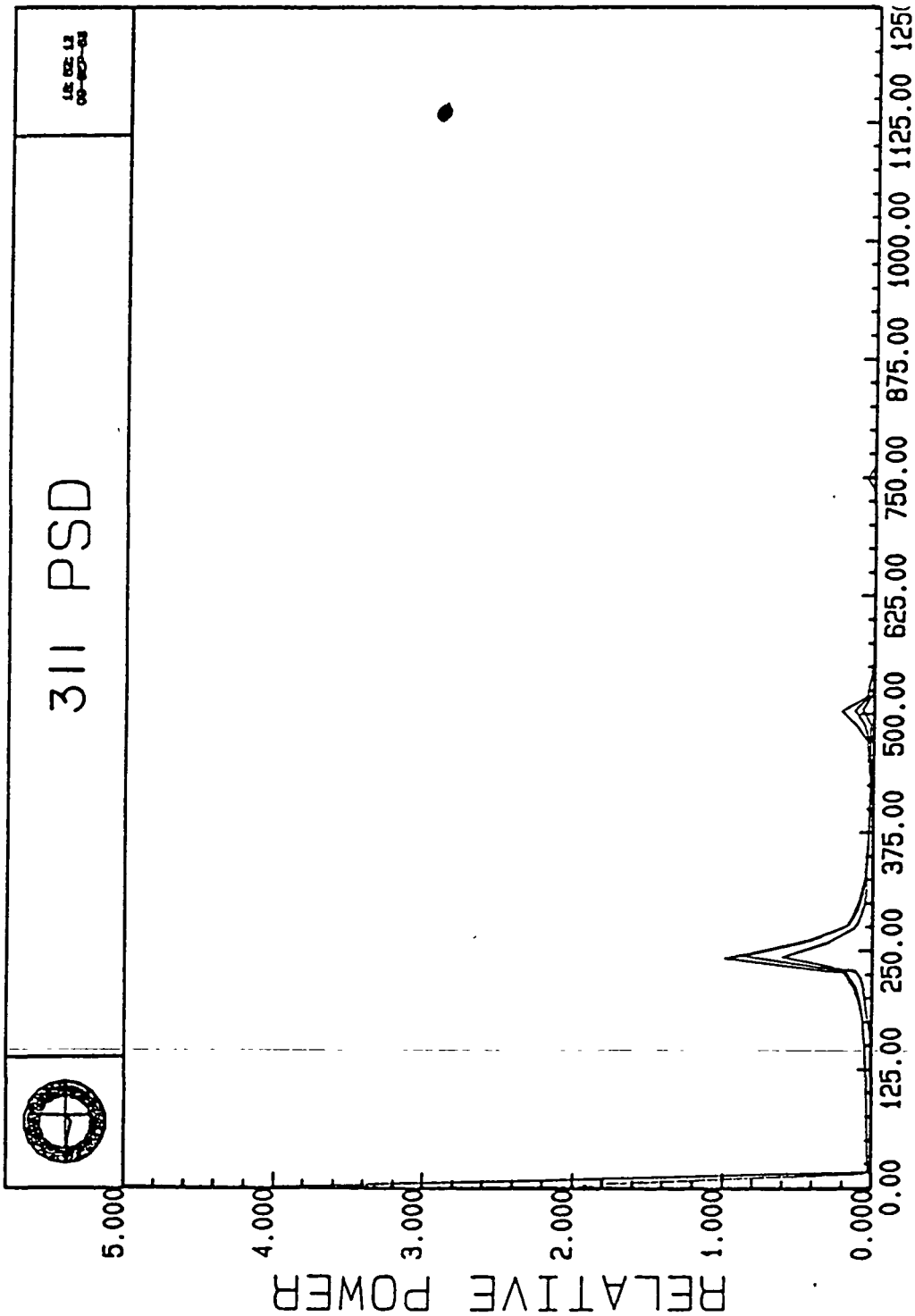

Figure 2.9b PSD for 2.9a; Shaft Speed: 333 Hz.

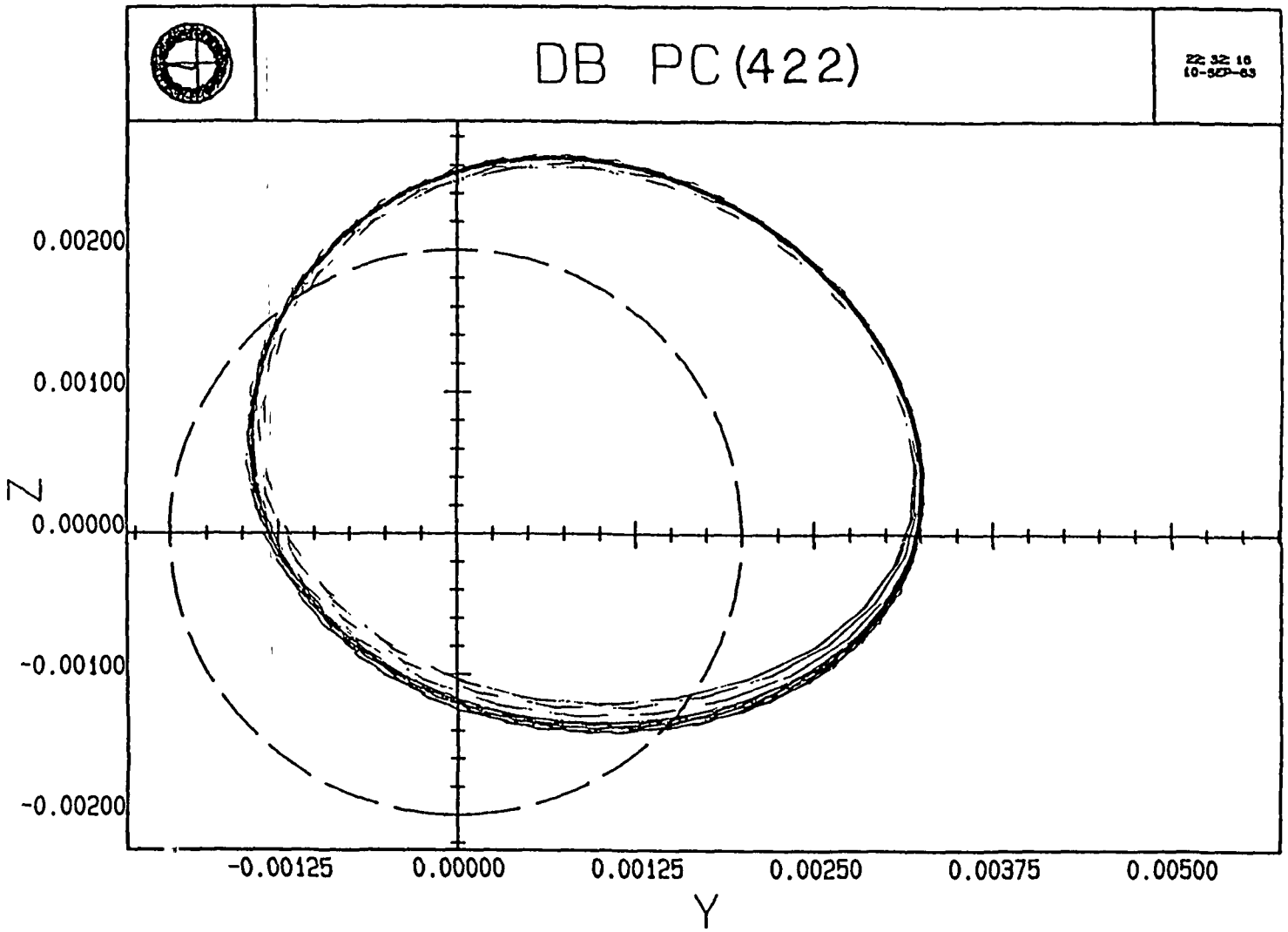

Figure 2.10a B-Type Motion Surrounding Origin and Part of the Deadband.

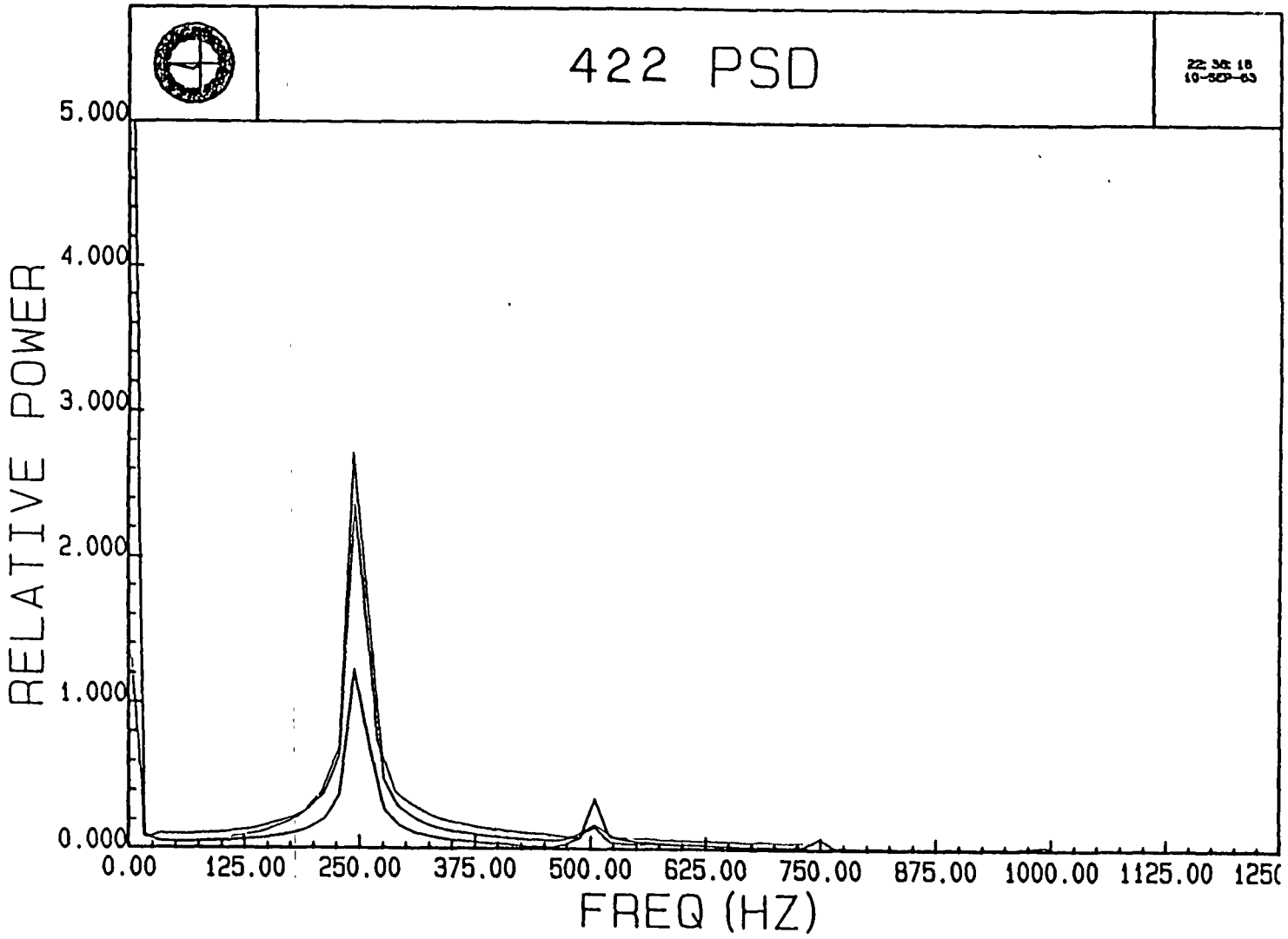

Figure 2.10b PSD for 2.10a; Shaft Speed: 500 Hz.

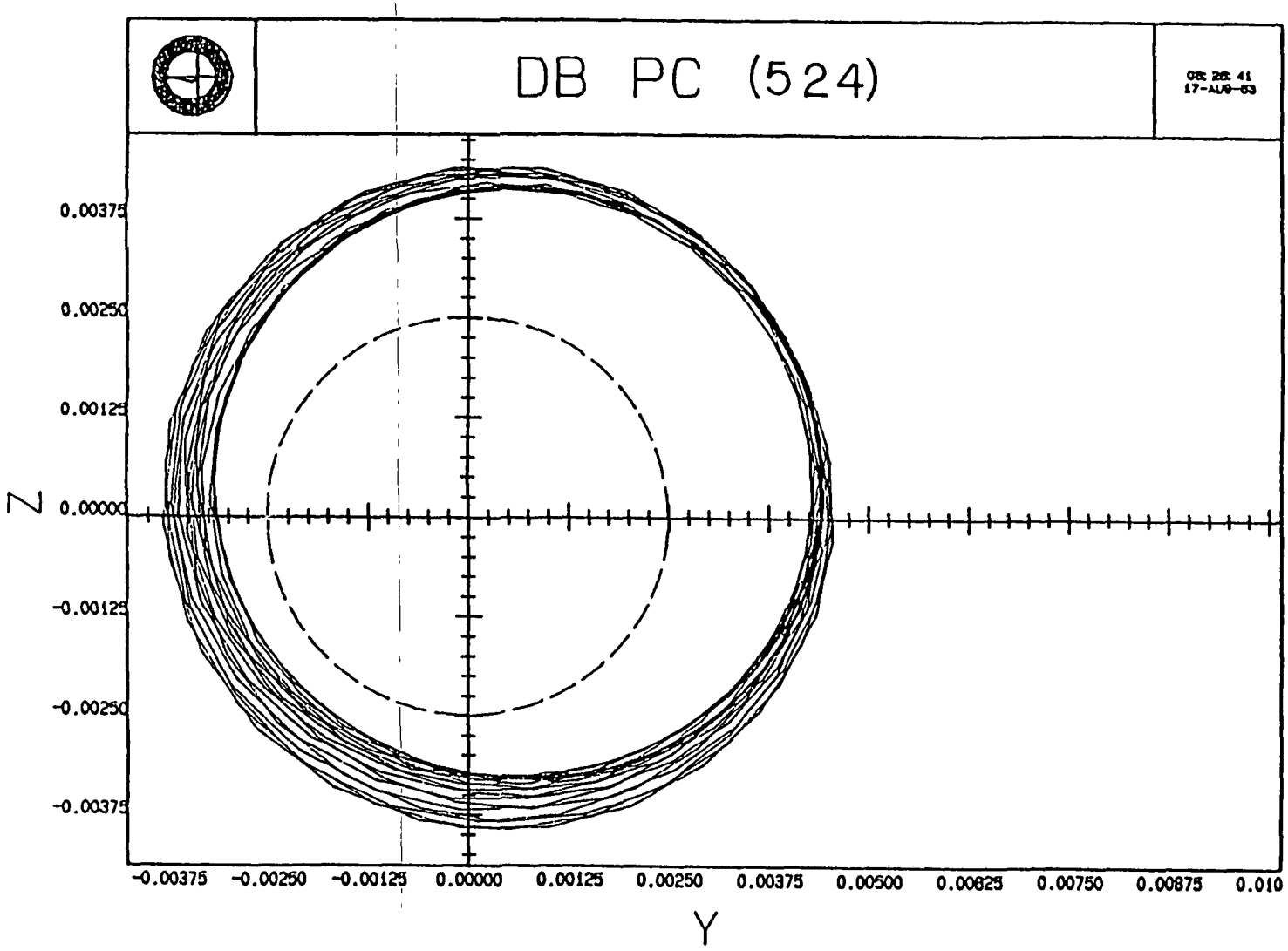

Figure 2.11a B-Type Motion Surrounding Entire Deadband.

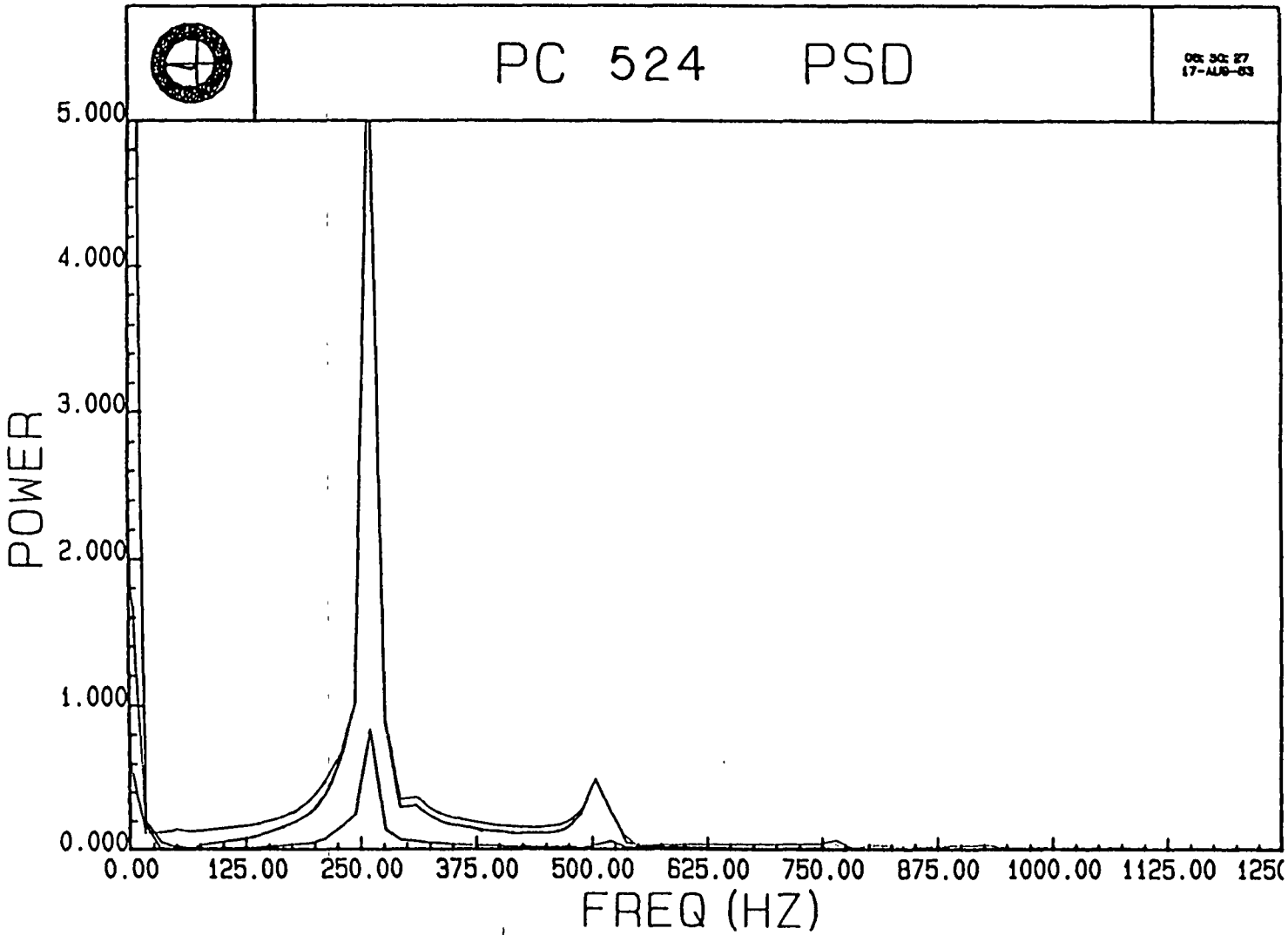

Figure 2.11b PSD for 2.11a; Shaft Speed: 500 Hz.

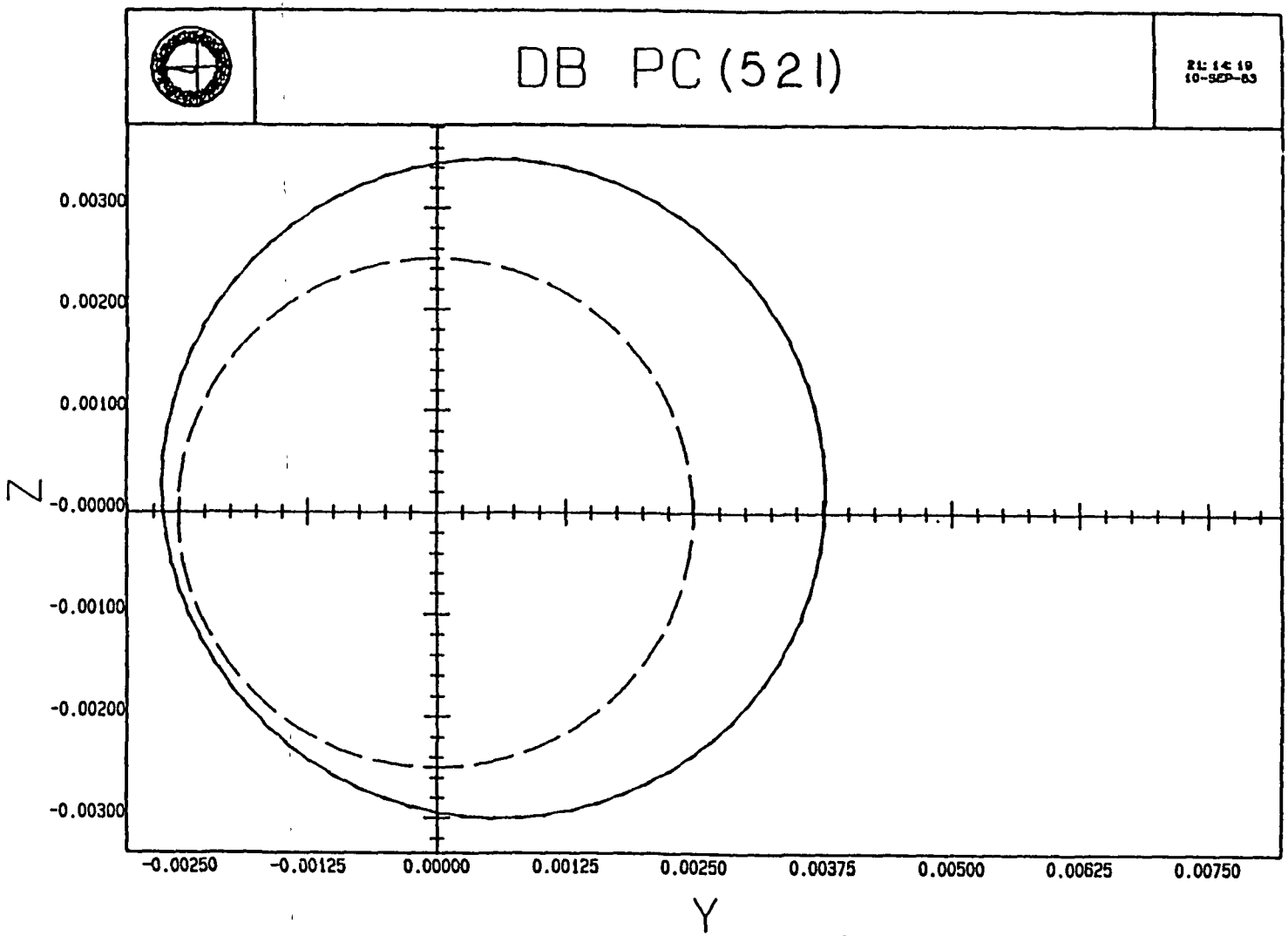

Figure 2.12a C-Type Motion.

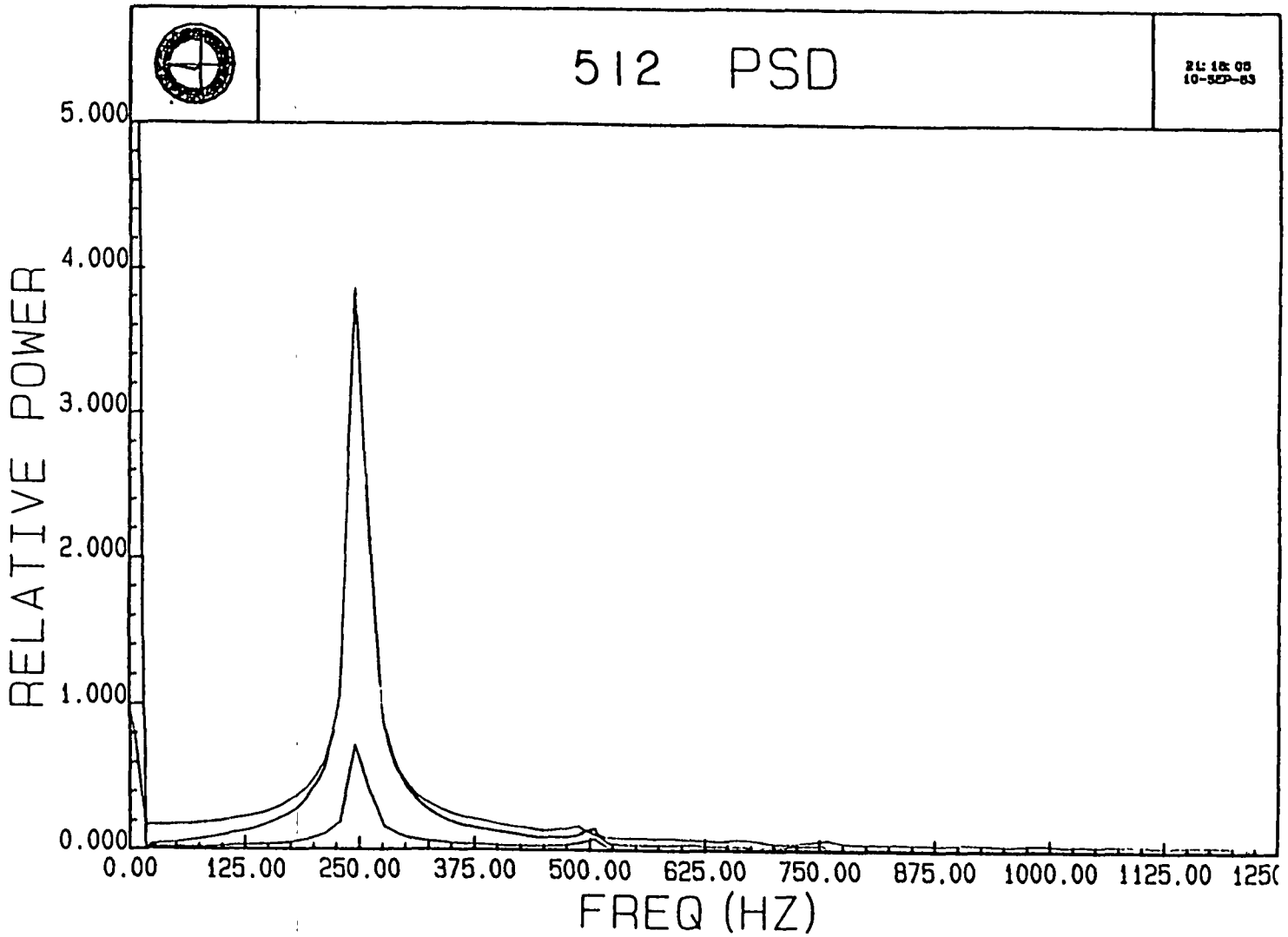

Figure 2.12b PSD for 2.12a; Shaft Speed: 500 Hz.

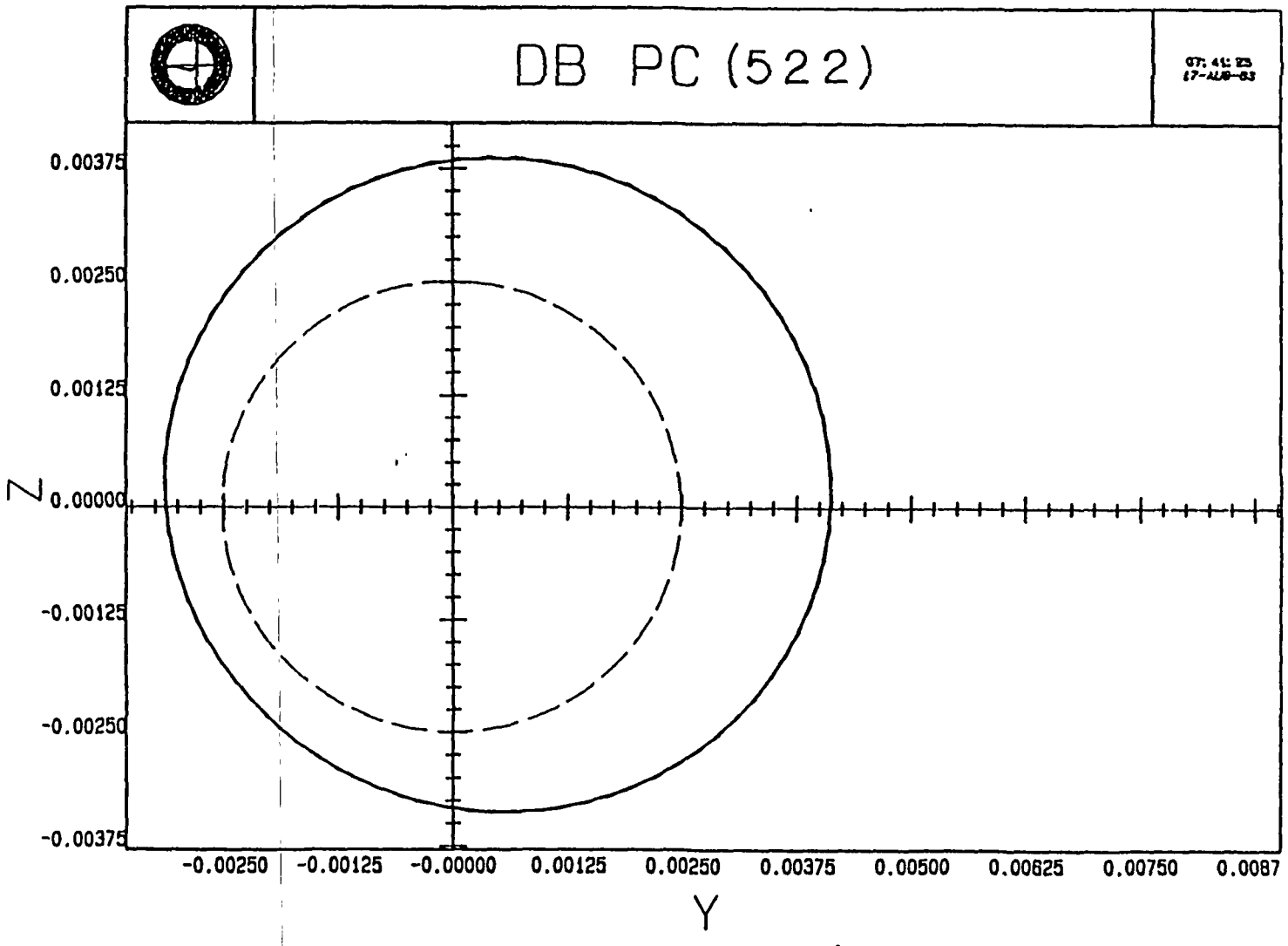

Figure 2.13a C-Type Motion.

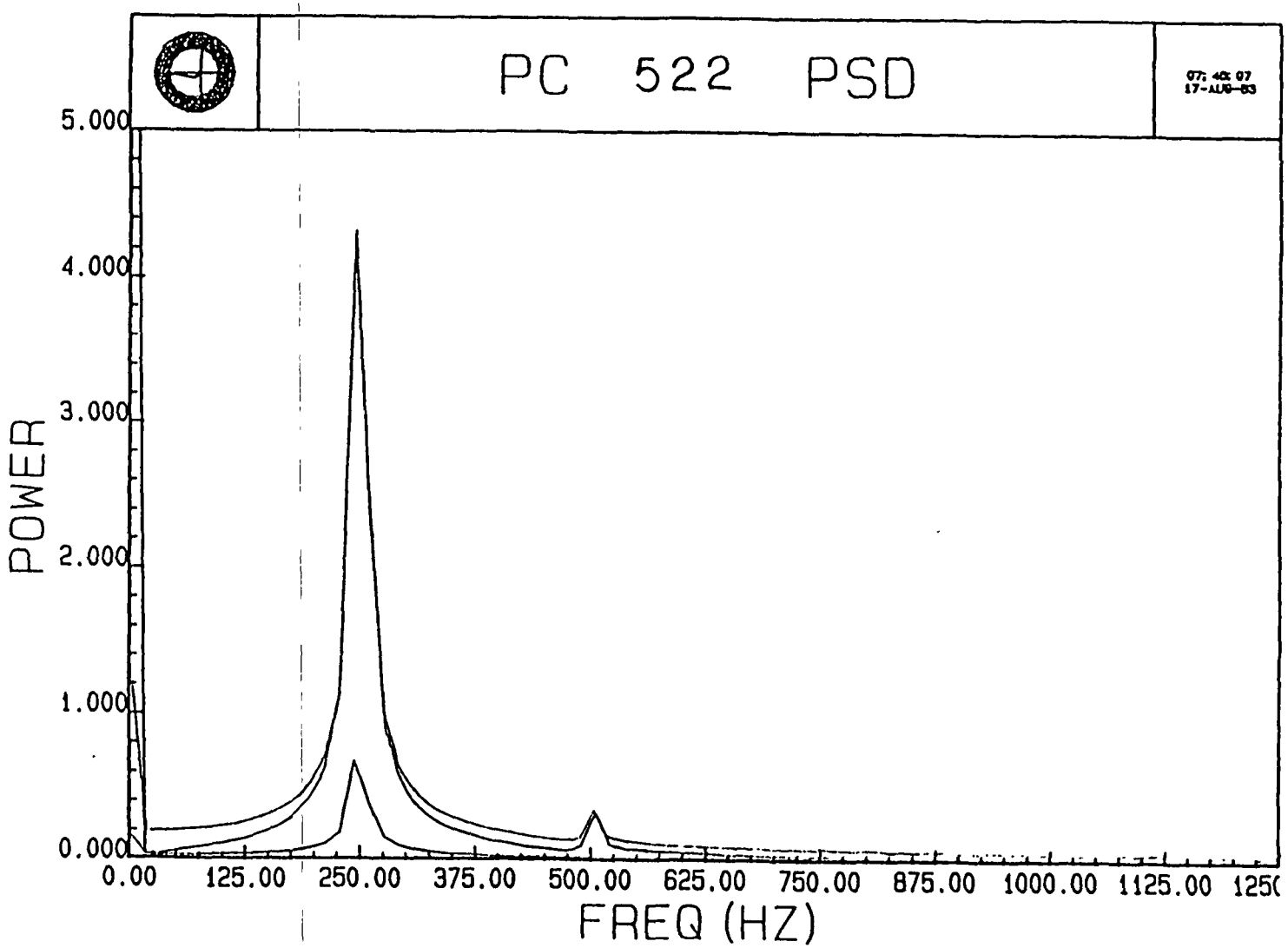

Figure 2.13b PSD for 2.13a; Shaft Speed: 500 Hz.

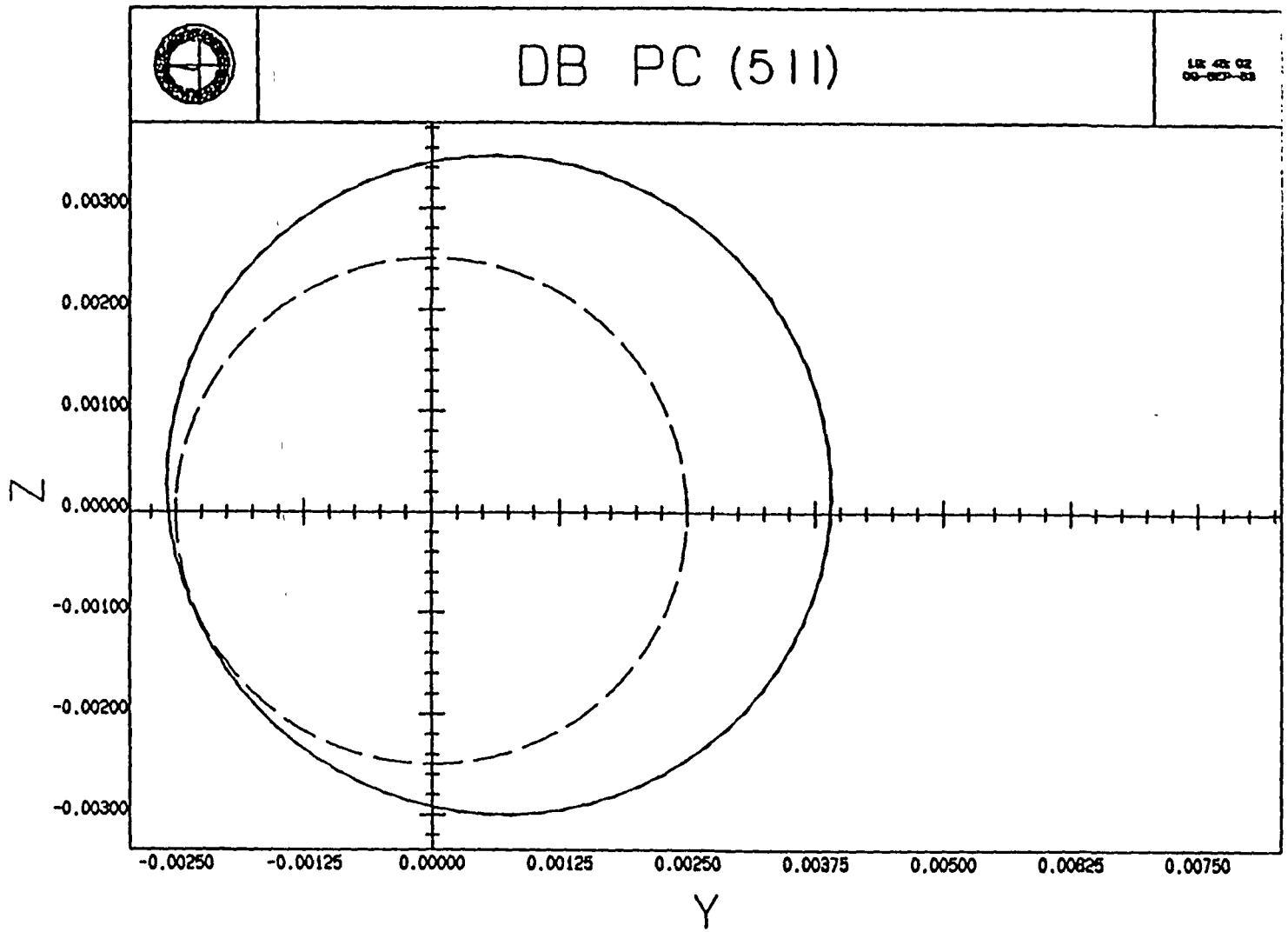

Figure 2.14a C-Type Motion Touching Deadband.

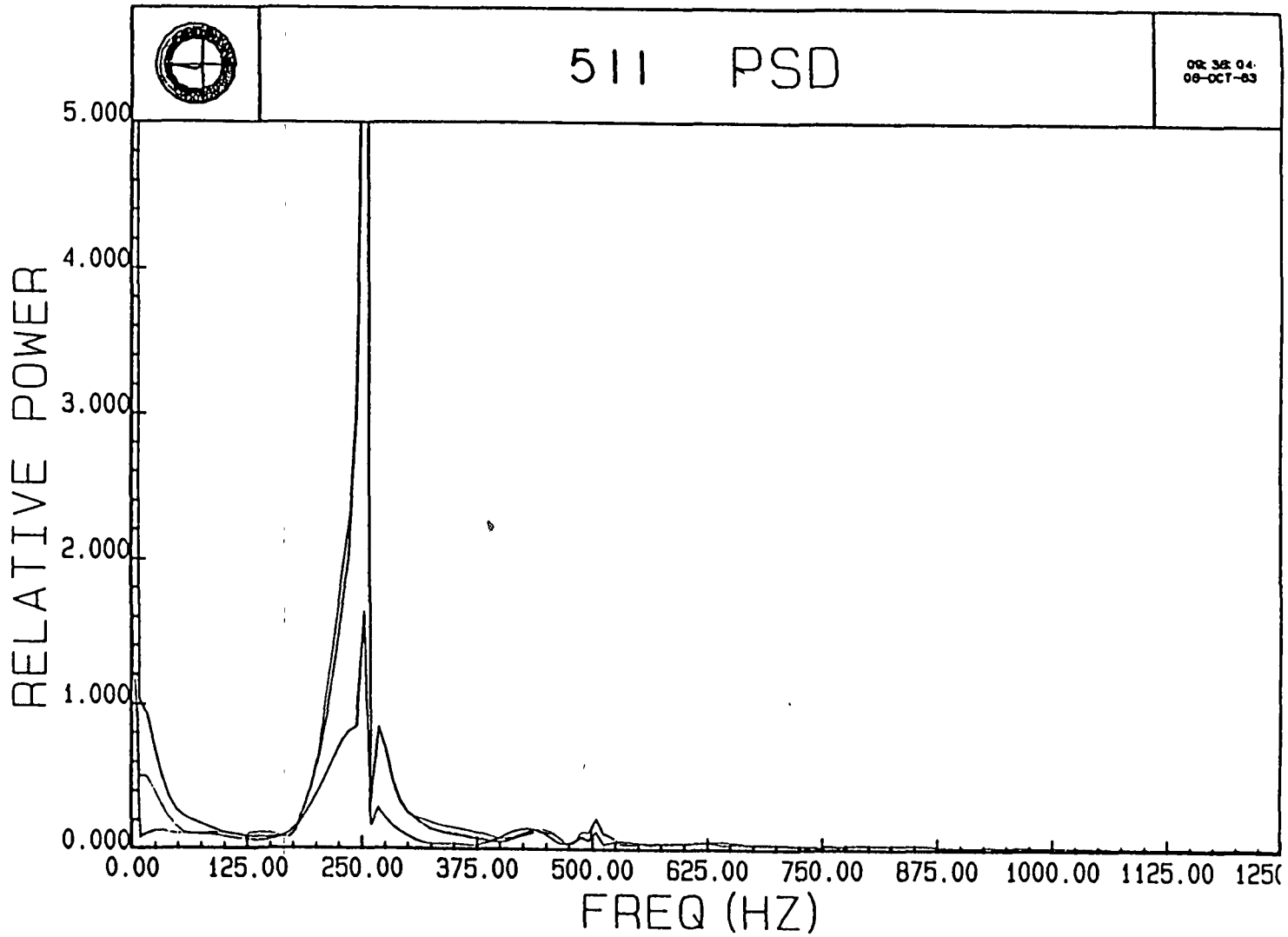

Figure 2.14b PSD for 2.14a; Shaft Speed: 500 Hz.

The procedure employed to study the various orbit types is to vary the shaft speed, rotor imbalance, deadband, and side force magnitude and subsequently execute the simulation. The other parameters in the system are assumed to remain constant and are listed in Table 2.1 below. A relevant note is that the integration step size used in the simulation is 30 microseconds. System steady-state is generally reached in somewhat less than 0.20 seconds, simulation time.

TABLE 2.1
VALUES OF CONSTANT SYSTEM PARAMETERS

PARAMETER	DESCRIPTION	VALUE
K_B	Bearing Stiffness	10^6 lbs/in
K_S	Seal Stiffness	2.0×10^5 lbs/in
C_S	Seal Damping Coefficient	200 lbs-sec/in
C_Q	Cross Couple Damping	40 lbs-sec/in
Q_S	Cross Couple Stiffness	$\frac{C_{S\omega}}{2}$ lbs/in
m	Rotor Mass	0.20422 lbs-sec^2/in

In order to keep the number of simulation runs to a minimum, only two values of shaft spin are considered. These are a rate of 2094.4 radians/second (333 Hz) and 3141.59 radians/second (500 Hz). Also, two values of rotor imbalance are used, 0.0001 and 0.0002 inches. Three side force values are used, 600, 800, and 1200 pounds in the positive y direction. Because the emphasis of the study is on the effects of deadband, five deadband values are considered, 0.0005, 0.001, 0.0015, 0.002, and 0.0025 inches. The values of the system parameters in Table 2.1, those of the shaft spin, shaft eccentricity, side forces, and deadbands are representative of those characteristic to the space shuttle high-pressure oxygen turbopump. To simplify the run identification process, a numbering convention is used to indicate the deadband, imbalance, and side force values in each case. The key to the numbering convention is given in Table 2.2. The results of the orbital study simulation runs are tabulated in Tables 2.3 and 2.4. Note that the frequency content of the orbits is also indicated. Only the two most prominent frequencies are indicated.

TABLE 2.2
KEY TO THREE DIGIT RUN IDENTIFICATION CODE

First Digit Deadband	Second Digit Imbalance	Third Digit Side Force
1 - 0.0005	1 - 0.0001	1 - 600
2 - 0.001	2 - 0.0002	2 - 800
3 - 0.0015		3 - 1000
4 - 0.002		4 - 1200
5 - 0.0025		

TABLE 2.3
ORBITAL TYPES - SHAFT SPEED OF 2094.4 RADIANS/SECOND (333 HZ)

RUN I.D.	TYPE ORBIT	FREQUENCY CONTENT(HZ) 1°	2°	COMMENTS
111	A	333	666	elliptical orbit outside deadband
211	A	333	-	elliptical orbit outside deadband
311	A	333	665	elliptical orbit ~ 1/3 inside deadband
411	A	333	-	elliptical orbit ~ 2/3 inside deadband
511	B	333	167	orbit within deadband
112	A	333	666	elongated elipse outside deadband
212	A	333	666	elongated elipse outside deadband
312	A	333	-	elongated elipse outside deadband
412	A	333	-	ellipse passing slightly inside deadband
512	A	333	666	ellipse ~ 1/2 inside deadband
114	A	333	666	ellipse outside deadband
214	A	333	666	ellipse outside deadband
314	A	333		ellipse outside deadband
414	A	333	-	ellipse outside deadband
514	A	333	-	ellipse outside deadband
121	A	333	666	large ellipse 1/5 inside deadband
221	A	333	666	elliptical orbit ~ 1/3 inside deadband
321	A	333	666	ellipse ~ 1/2 inside deadband
421	A	333	666	ellipse ~ 2/3 inside deadband
521	B	333	167	orbit totally inside deadband
122	A	333	167	large ellipse passing slightly within deadband
222	A	333	666	large ellipse passing slightly within deadband
322	A	333	666	ellipse ~ 1/4 inside deadband
422	A	333	666	ellipse ~ 1/3 inside deadband
522	A	333	666	ellipse ~ 1/2 inside deadband
124	A	333	666	ellipse totally outside deadband
224	A	333	666	ellipse totally outside deadband
344	A	333	666	ellipse totally outside deadband
424	A	333	666	ellipse almost touching deadband
524	A	333	666	ellipse passing slightly within deadband

TABLE 2.4
ORBITAL TYPES - SHAFT SPEED OF 3141.59 RADIANS/SECOND (500 HZ)

RUN I.D.	TYPE ORBIT	FREQUENCY CONTENT(HZ) 1°	2°	COMMENTS
111	A	500		ellipse outside deadband
211	A	500	500	ellipise passing slightly inside deadband
311	B	250	500	orbit ~ 1/2 inside deadband
411	B	250	500	large orbit surrounding deadband
511	C	250	-	circular orbit about deadband
112	A	500	-	ellipse outside deadband
212	A	500	-	ellipse outside deadband
312	A	500	-	ellipse just touching deadband
412	B	250	500	
512	B	250	500	orbit surrounds deadband
114	A	500	-	ellipse outside deadband
214	A	500	-	ellipse outside deadband
314	A	500		ellipse outside deadband
414	A	500	-	ellipse outside deadband
514	A	500	-	ellipse outside deadband
121	A	500	1000	large ellipse passing slightly into dead-band
221	A	500	1000	large ellipse ~ 1/4 inside deadband
321	B	250	500	overlaps most of deadband
421	C	250	500	encircles most of deadband
521	C	250	500	totally encircles deadband
122	A	500	1000	large ellipse outside deadband
222	A	500	1000	large ellipse passing slightly inside band
322	A	500	-	ellipse ~ 1/5 within deadband
422	B	250	500	encircles most of deadband
522	C	250	500	circular orbit encircling deadband
124	A	500	-	ellipse outside deadband
224	A	500	-	ellipse outside deadband
344	A	500	-	ellipse outside deadband
424	A	500	-	ellipse just touching deadband
524	B	250	500	totally surrounds deadband

The apparent trend is that with smaller deadbands, the orbits tend to be of type A and at the synchronous frequency. As the deadband increases, the orbit transitions to a B-type motion with several frequencies present in the PSD. Finally, at the larger deadband values, the orbit becomes type C at half synchrounous frequency. This tendency is illustrated with the PSD plots shown in Figures 2.15 and 2.16. Figure 2.15 shows PSD plots for runs made with no side forces acting in the system. As the deadband is increased from zero to 0.0025 inches, the frequency content of the resulting orbit changes from synchronous only to a combination of synchronous and halfsynchronous frequencies, with the halfsynchronous frequency dominating in magnitude. Figure 2.16 illustrates the point further. These PSD plots were generated from runs made with 600 pounds of side force acting on the system. Again, as the deadband is increased from zero to 0.0025 inches, the frequency content of the resulting orbit changes from synchronous to predominently half-synchronous as the orbit type changes from A to C-type motion.

For the cases which were run with a shaft speed of 2094.4 radius/second, the orbits tended to be A-type even for the larger deadbands. The presence of a side force apparently suppresses a C-type whirl as can be seen by examining table 4, specifically runs 114 through 514. The higher side force case showed no B or C type motion for the smaller imbalance term. The effect of whirl orbits on bearing loads will be discussed in some detail in a later section of this report.

2.3 LIMIT CYCLE ANALYSIS

Examination of the whirl orbits presented in the plots in the previous section indicate that for A and C-type orbits, the motion of the rotor is a limit cycle. All four of the system states, r, $\dot{r}$, ϕ, and $\dot{\phi}$ are periodic in time for the A-type motion. For the case of C-type motion, this periodicity is also readily apparent for three of the states; however, since the orbit encircles the origin, the magnitude of the whirl angle, ϕ, is always increasing.

To characterize the limit cycle motions present in the rotor system, an algorithm has been developed which will converge to a set of initial conditions for the four system states which, when input into the simulation, will cause the system to immediately exhibit the limit cycle behavior. The algorithm has been implemented in the form of a computer program. The methods it employs to find these initial conditions are explained below.

The algorithm is based on the fact that the function for which the limit cycle initial conditions are sought is periodic. That is, the orbit comes back around to the same point once each cycle. The idea is to determine that such a point exists and the values of the system states which satisfy this condition.

Given the state equations which describe the system, a solution to the states may be obtained through integration. The mathematical statement of the problem is:

$$\dot{\underline{p}} = \underline{f}\,(\underline{p},t) \qquad (2.34)$$

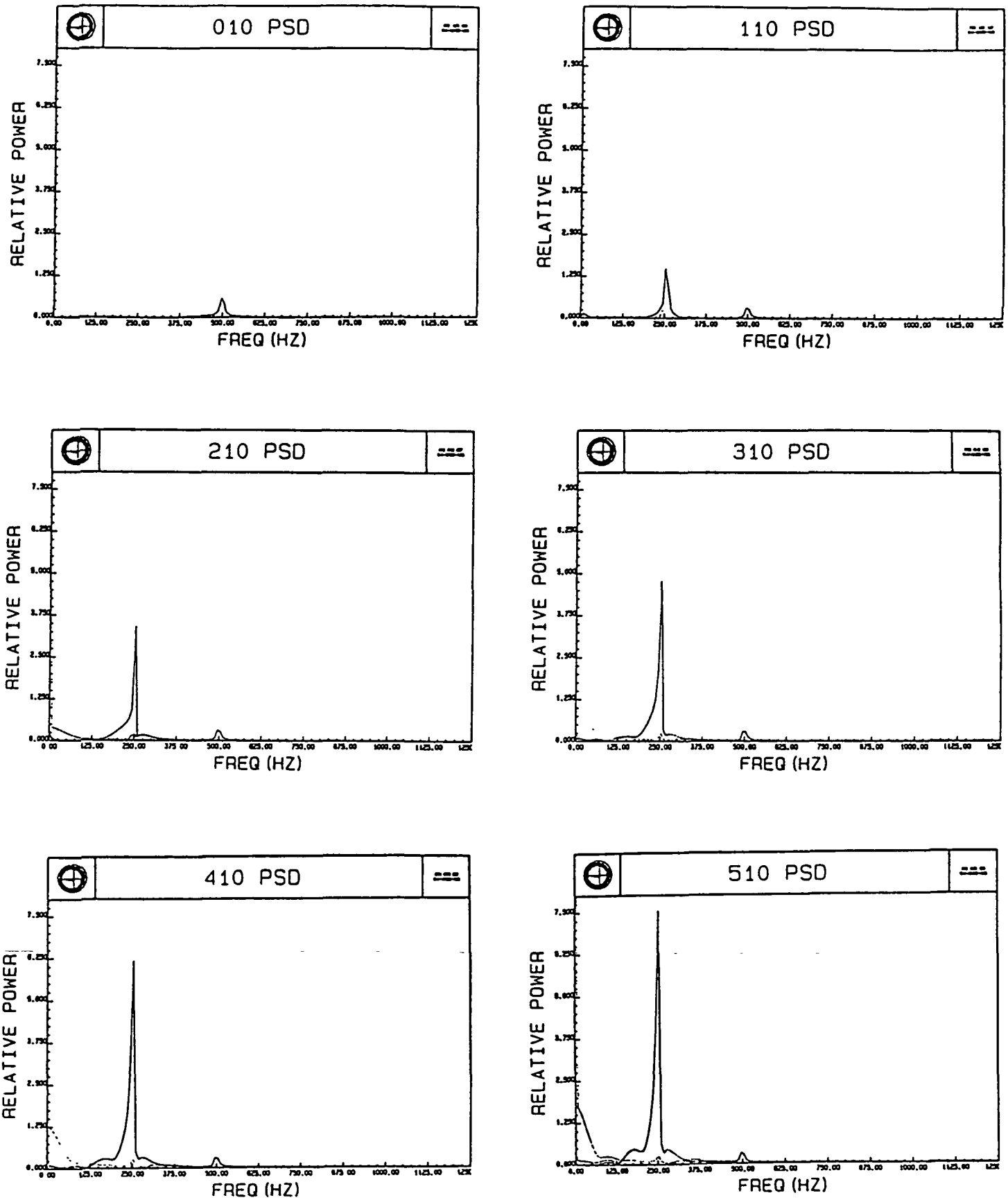

Figure 2.15 PSD Plots Indicating Frequency Content Change as Deadband Increases, the Side Force is Zero.

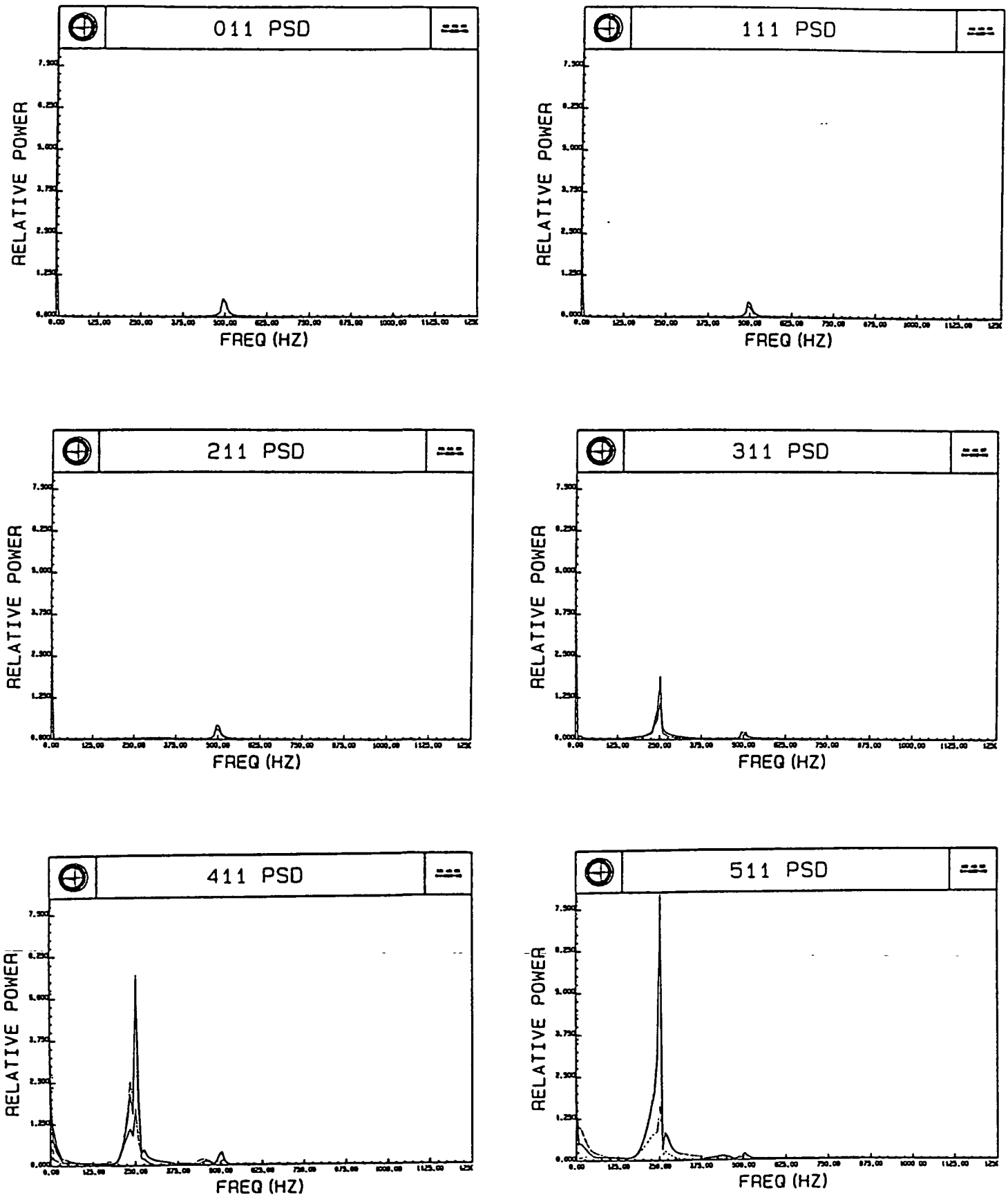

Figure 2.16 PSD Plots Indicating Frequency Content Changes as the Deadband Increases for a Side Force of 600 lbs.

the solution to which is

$$\underline{p}(t) = \underline{p}_o + \int_0^t \underline{f}\,(\underline{p},\tau)d\tau \tag{2.35}$$

where $\underline{p}_o$ is some initial state. It is desired to determine the $\underline{p}_o$ such that

$$\underline{p}(t) = \underline{p}_o \quad \text{for } t = T \tag{2.36}$$

$$\text{where } T = \frac{2\pi}{\omega} \tag{2.37}$$

the period of the function. In other words, we wish to determine $\underline{p}_o$ so that the integral in Equation 2.34 is zero. The problem may be restated as

$$\underline{p}(T) = \underline{p}_o + \underline{g}(\underline{p}) \ . \tag{2.38}$$

$$\text{If } \underline{g}(\underline{p}) \text{ can be driven to zero, then } \underline{p}(T) = \underline{p}_o \ . \tag{2.39}$$

The function $\underline{g}(\underline{p})$ may be approximated to 1st order by

$$\underline{g}(\underline{p}) = \underline{g}\,(\underline{p}_o) + \left[\frac{\partial \underline{g}}{\partial \underline{p}}\right]_{\underline{p}_o} \cdot \Delta\,\underline{p} \tag{2.39}$$

where $\Delta\underline{p}$ is some incremental change in the state vector $\underline{p}$. This is the quantity to be determined. It will be added to the original state vector. Because we wish $\underline{g}(\underline{p})$ to be zero, Equation 2.39 is rewritten as

$$\underline{0} = \underline{g}\,(\underline{p}_o) + \underline{\underline{J}} \cdot (\underline{p} - \underline{p}_o) \tag{2.40}$$

where

$$\Delta\underline{p} = \underline{p} - \underline{p}_o \tag{2.41}$$

and $\underline{\underline{J}}$ is the Jacobian of $\underline{g}(\underline{p})$. The solution for $\Delta\underline{p}$ is

$$\Delta\underline{p} = \underline{\underline{J}}^{-1}\,(-\underline{g}(\underline{p}_o)) \ . \tag{2.42}$$

The way the computer implementation of the algorithm works is that an initial state vector $\underline{p}_o$ is input to the routine. Using the same integration scheme as that used in the rotor model simulation, the value of the state vector at time $t = T$ is determined. The value of $\underline{g}(\underline{p}_o)$ is then determined by

$$\underline{g}\,(\underline{p}_o) = \underline{p}\,(T) - \underline{p}_o \tag{2.43}$$

Numerically, the partial derivative of the function $\underline{g}\,(\underline{p})$ may be approximated by

$$\frac{\partial \underline{g}}{\partial \underline{p}_i} \cong \frac{\underline{g}\,(\underline{p}_{oi} + \delta\underline{e}_i) - \underline{g}\,(\underline{p}_{oi})}{\delta} \tag{2.44}$$

where δ is some very small increment in the ith state and e_i is the unit vector in the direction of the ith state. The Jacobian for $\underline{g}(\underline{p})$ may thus be determined using this approximation for the partial derivatives. Once this matrix is formed, it is numerically inverted and multiplied by the negative of the vlaue of $\underline{g}(\underline{p}_o)$ to obtain $\Delta\underline{p}$. At this point a new set of initial states is formed as

$$\underline{p}o_{new} = \underline{p}o_{old} + \Delta\underline{p} \; . \tag{2.45}$$

The process is iterated until the values $\Delta\underline{p}$ and $\underline{g}(\underline{p})$ are within a specified tolerance. Once this occurs, the resulting state values are the initial conditions needed for the system to exhibit a limit cycle response.

Theoretically, if a limit cycle exists in the system, the algorithm described above should easily converge to the desired initial conditions. However, as with many numerical techniques, computational difficulties often arise. If the Jacobian matrix ever becomes singular, then the numerical method used to find the inverse does not work and the results from that point on are erroneous. To avoid this situation, logic checks are written into the code to prevent the Jacobian from becoming singular. Whenever this situation is detected, the state vector is reset and the process is restarted with the new state vector as input. It has been discovered that for this technique if the Jacobian continuously becomes singular, even after resetting the initial state vector, then convergence to the desired state will not be obtained and the algorithm is indeed diverging, and there is no solution.

For orbits which are C-type, another modification to the algorithm is required. Because the whirl angle, ϕ, is not periodic but increasing with time, this method left unmodified will not converge to a solution. To force ϕ to appear to be periodic, the value 2π is subtracted from the ϕ component of the state vector value at time $t = T$. This procedure will, in fact, allow the algorithm to converge to a solution to the C-type orbit initial conditions.

The algorithm has been thoroughly evaluated and found to converge to both A and C-type orbit initial conditions relatively quickly. Shown in Figure 2.17a is a plot of the orbit obtained in run 521 with a shaft speed of 333 Hz including all the transients. In Figure 2.17b is the plot of the resulting orbit obtained when the initial conditions from the convergence algorithm are input into the simulation. This demonstrates that, indeed, an initial state vector which immediately produces the limit cycle response has been determined. The PSD of the orbit is shown in Figure 2.17c. Another example of convergence to an A-type limit cycle is presented in Figure 2.18. Again the plot in 2.18a is the orbit including the transients when the rotor is started from the rest position. Figures 2.18b and 2.18c are the limit cycle orbits and the PSD plot respectively. The shaft speed for this case is 500 Hz.

Convergence to a C-type orbit is shown by the plots presented in Figure 2.19. This orbit is at the half-synchronous frequency of 250 Hz. This also must be taken into account when the convergence algorithm is used to determine an initial state vector because the procedure is quite sensitive to the period of the signal. Obviously, if the orbital frequency is 250 Hz, corresponding to a period of 0.004 seconds, if the period is input as 0.002, corresponding to an orbital frequency of 500 Hz, the algorithm will not converge. The point may seem trivial; however, its importance must be realized.

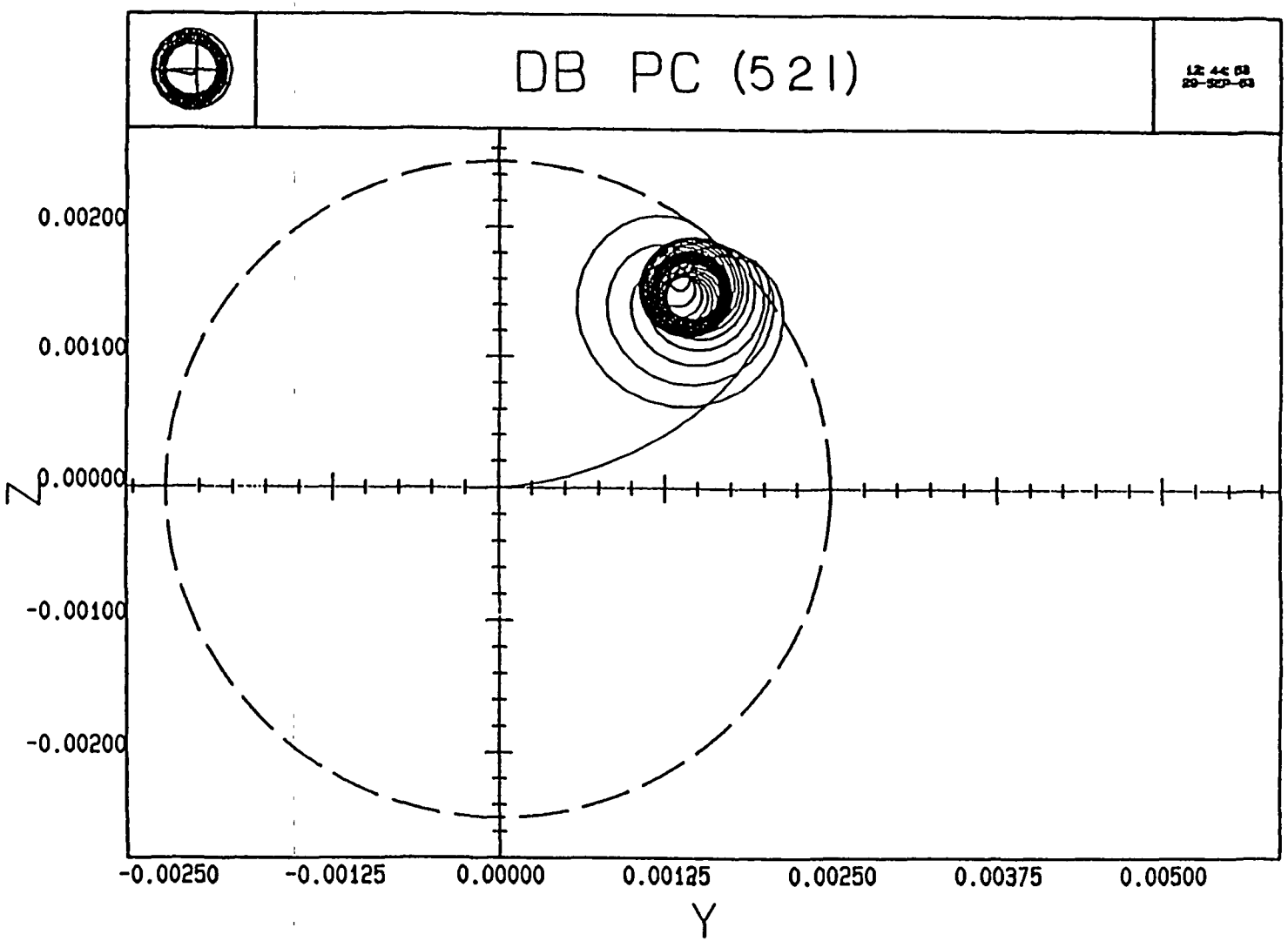

Figure 2.17a Plot of A-Type Limit Cycle Including Transients.

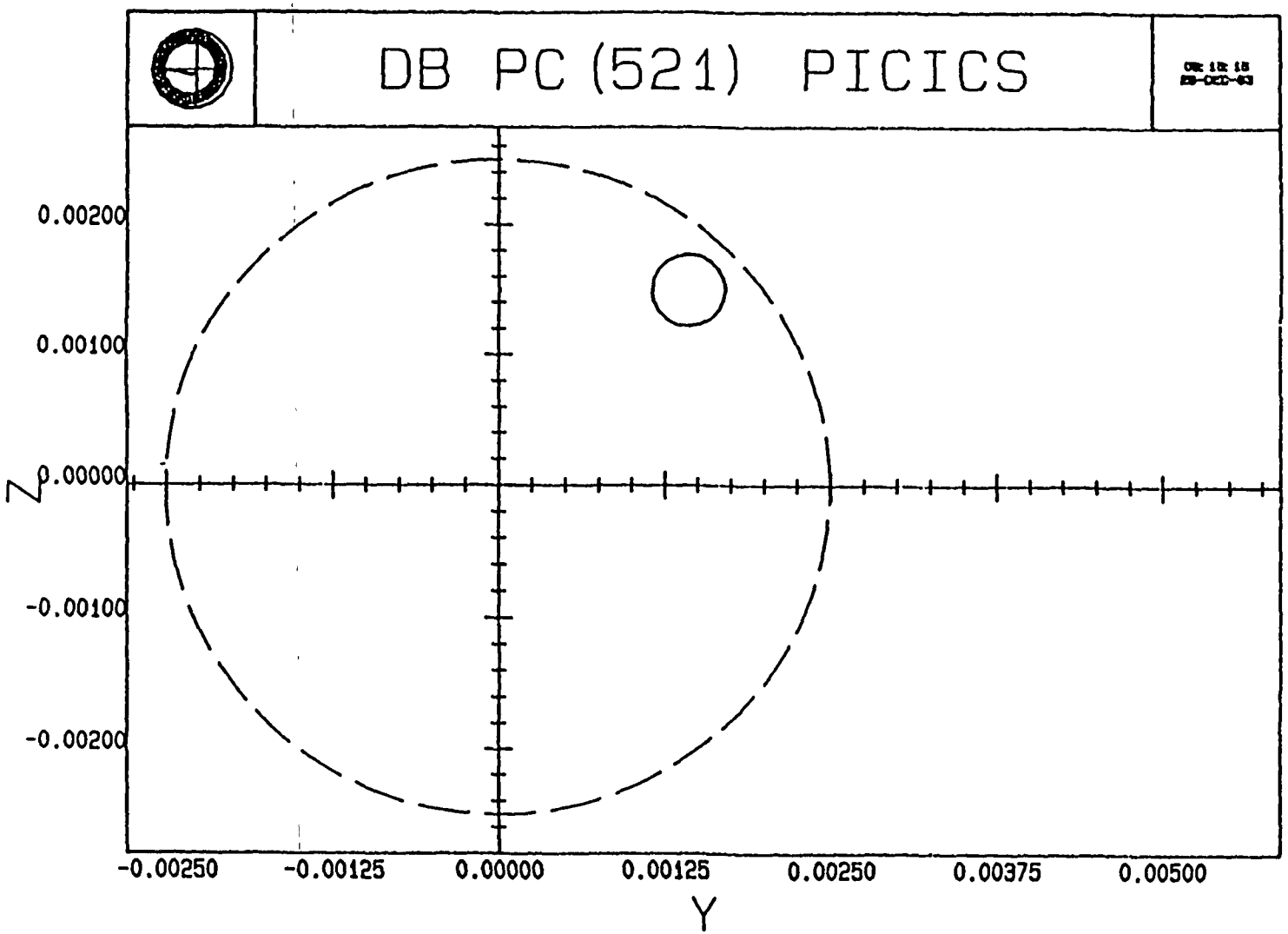

Figure 2.17b A-Type Limit Cycle from Convergence Algorithm Initial Conditions.

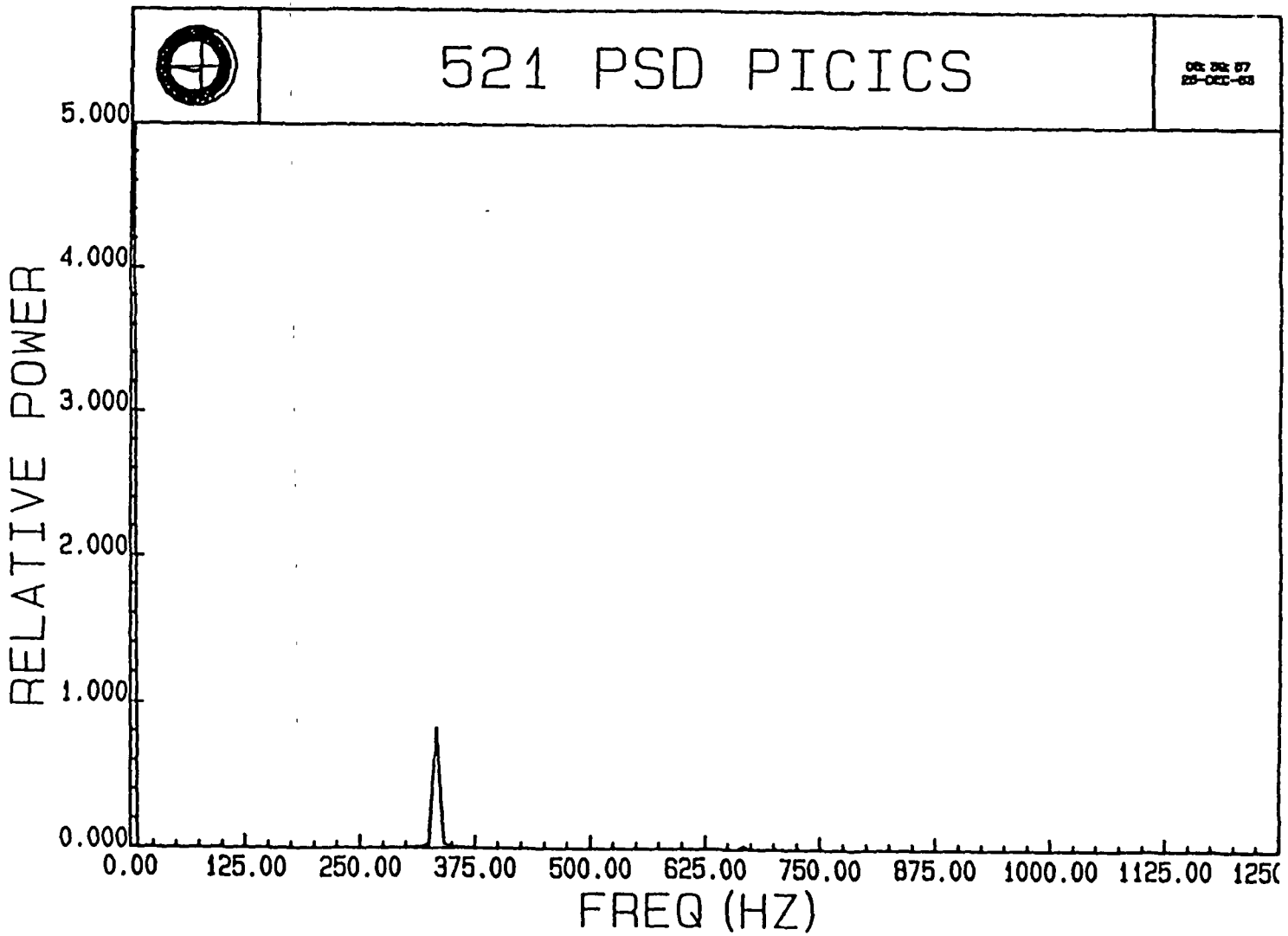

Figure 2.17c PSD for the Orbit in 2.17b; Shaft Speed: 333 Hz.

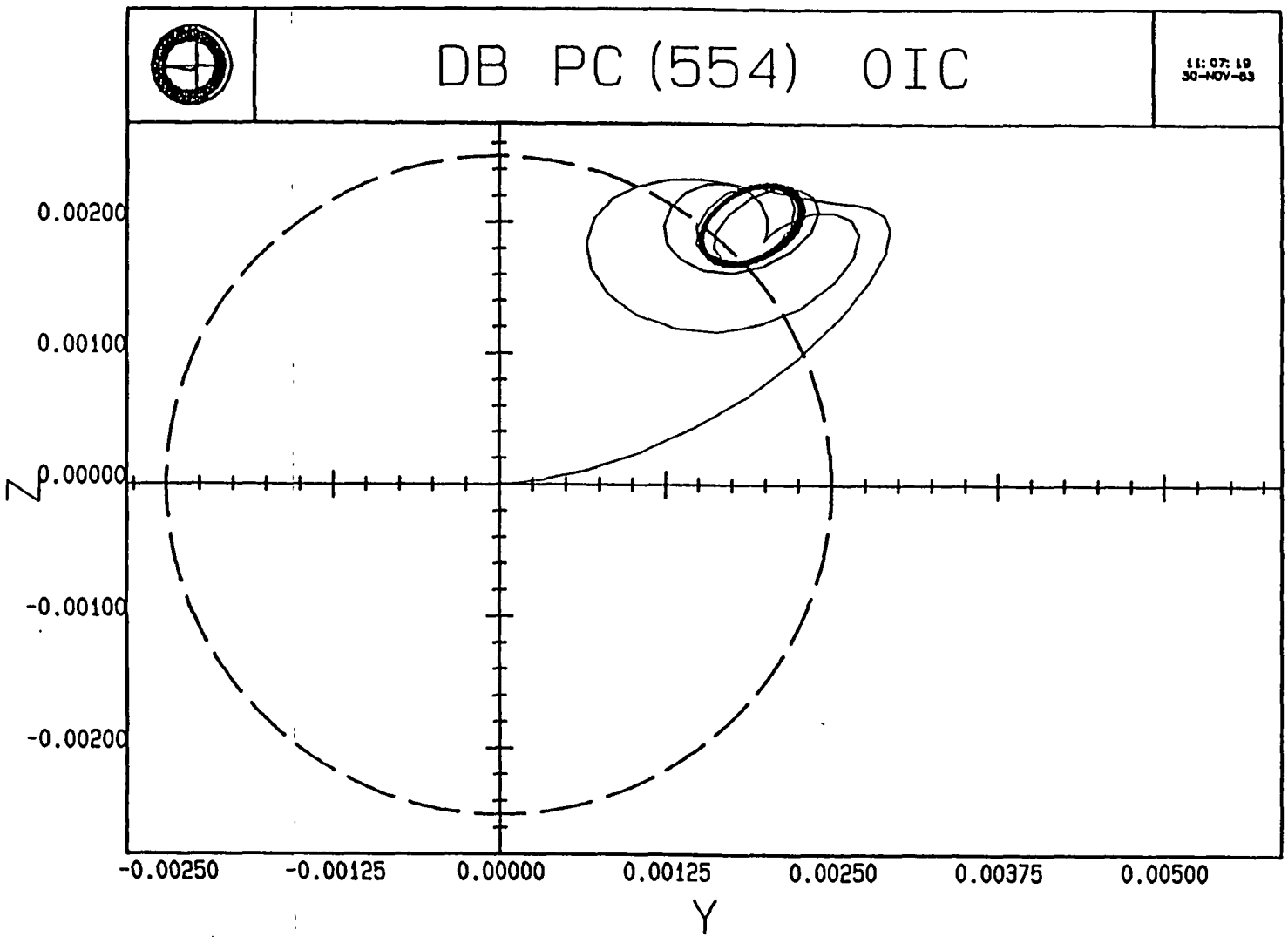

Figure 2.18a A-Type Limit Cycle Including Transients.

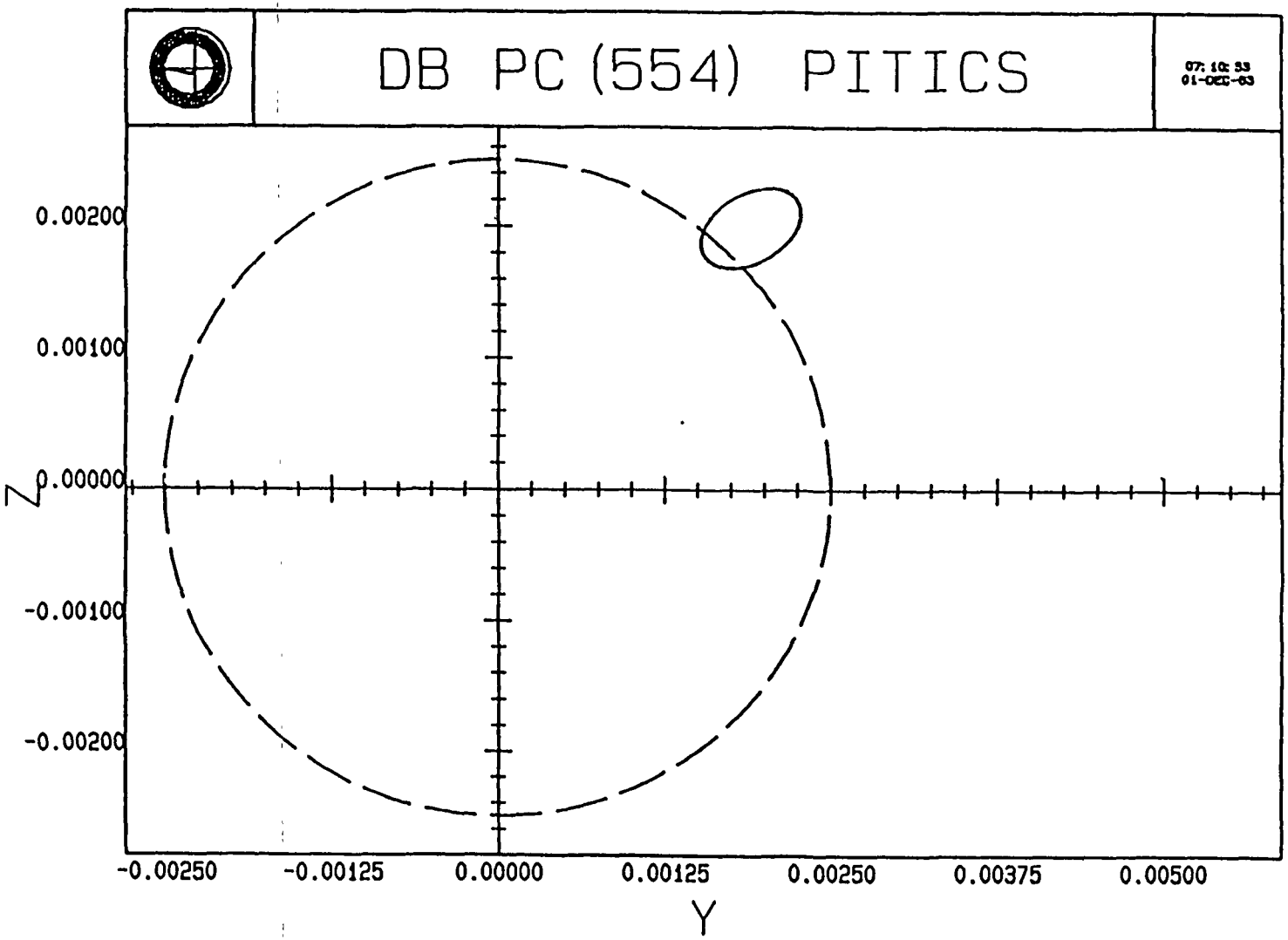

Figure 2.18b A-Type Limit Cycle from Convergence Algorithm Initial Conditions.

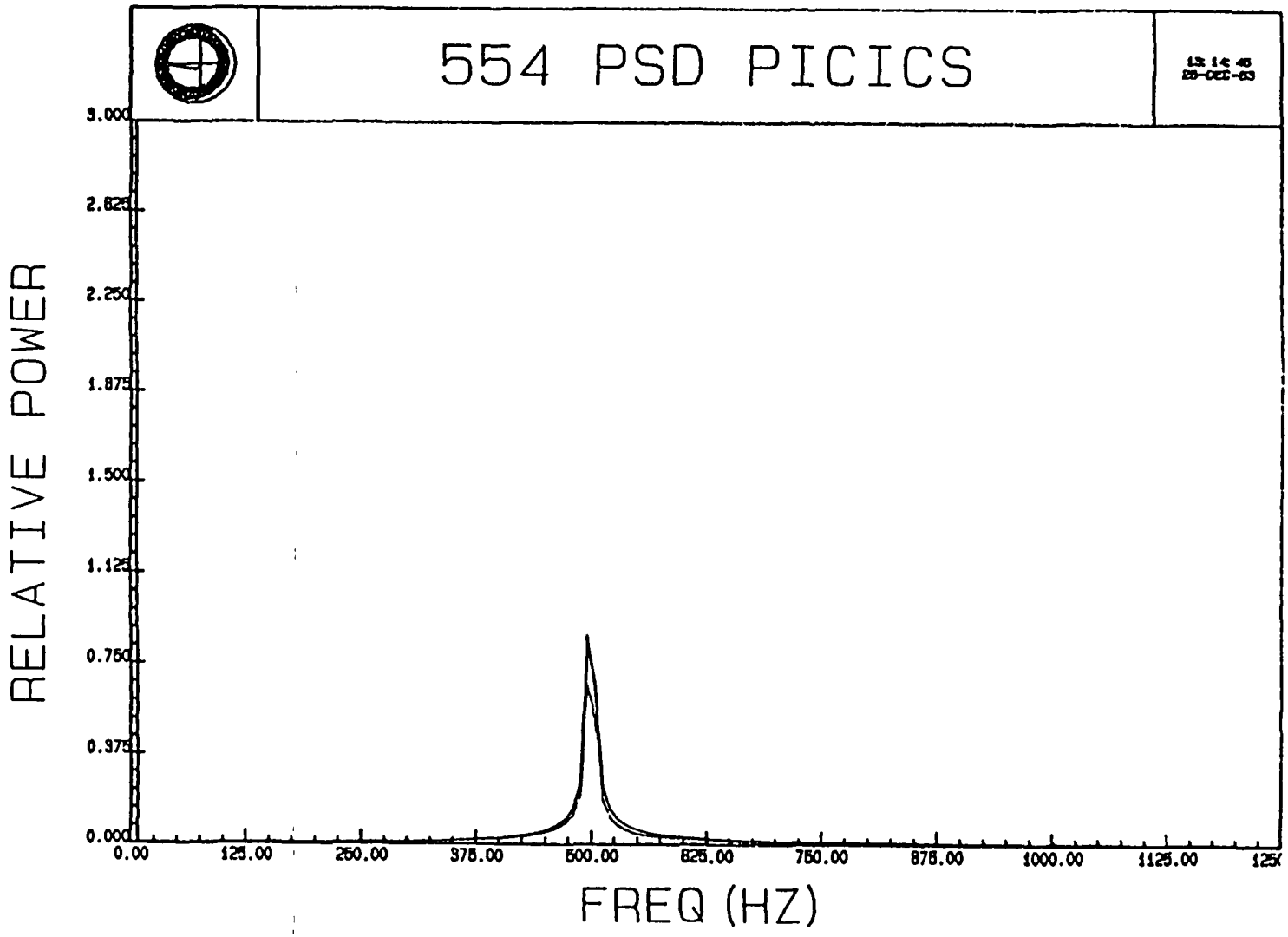

Figure 2.18c PSD for the Orbit in 2.18b; Shaft Speed: 500 Hz.

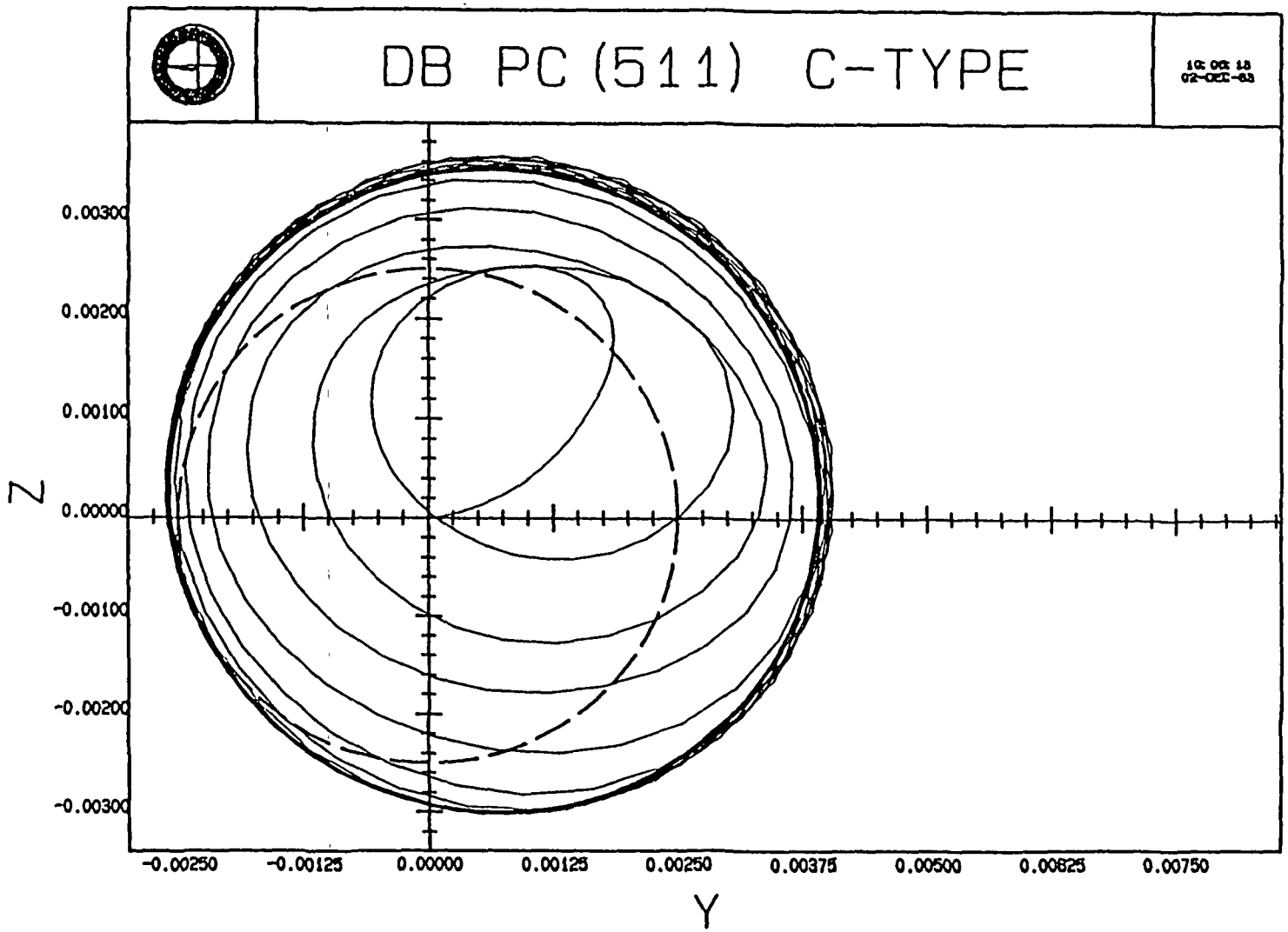

Figure 2.19a Subsynchronous C-Type Limit Cycle Including Transients.

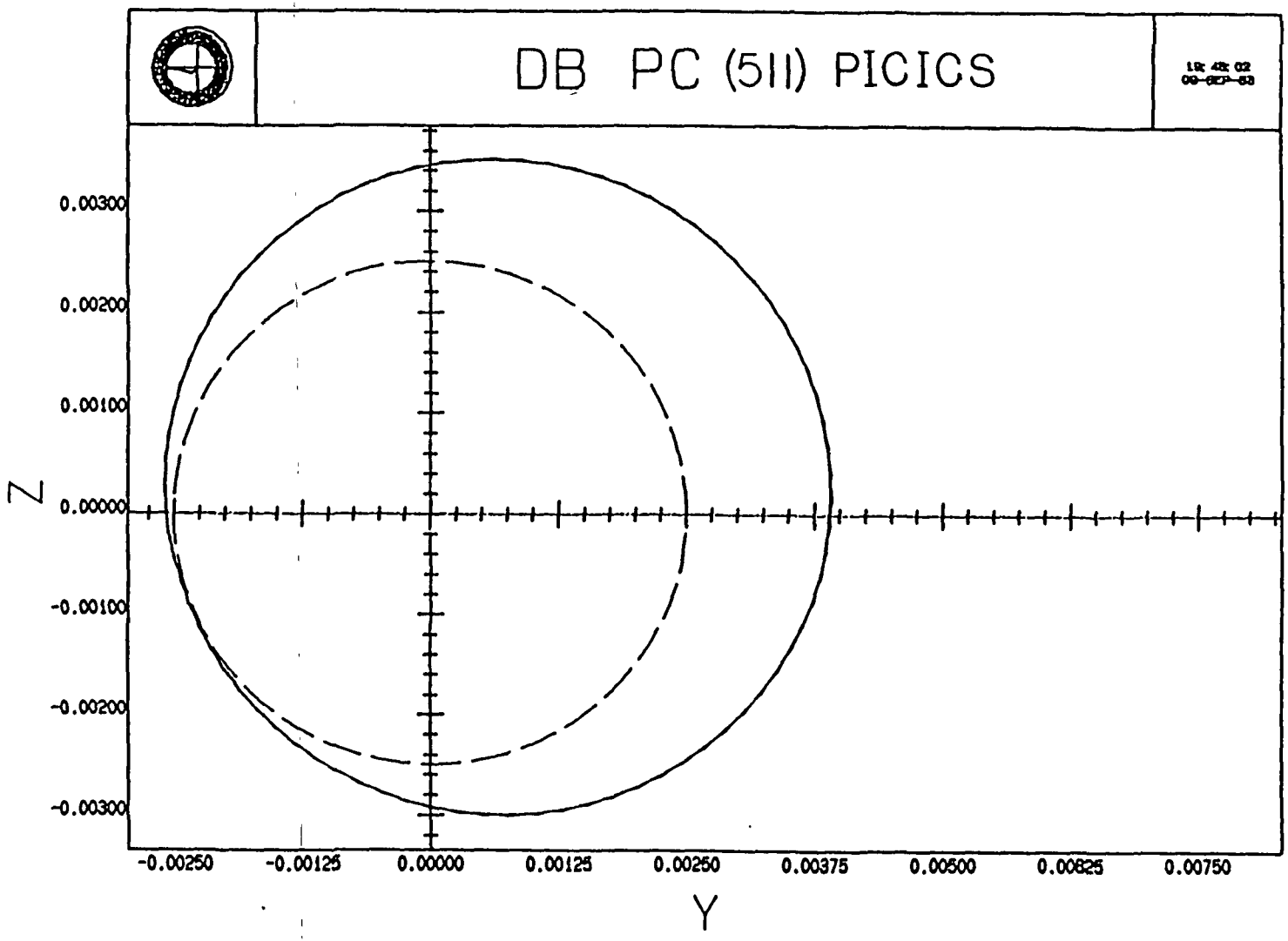

Figure 2.19b Subsynchronous C-Type Limit Cycle from Convergence Algorithm Initial Conditions.

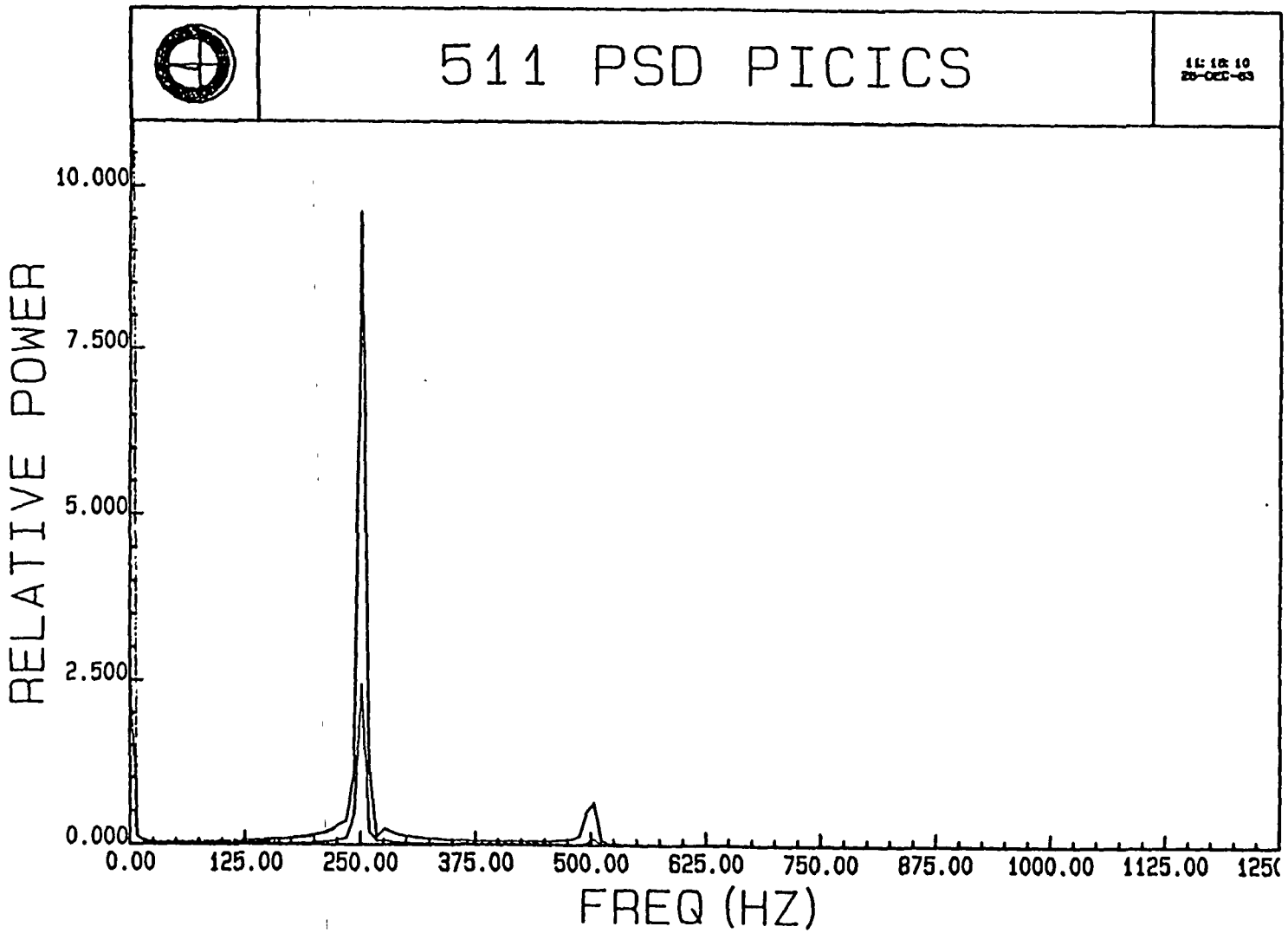

Figure 2.19c PSD for the Orbit in 2.19b; Shaft Speed: 500 Hz.

Convergence to a synchronous C-type orbit is also obtained with the algorithm. This type of orbit typically occurs when there is no side force present in the system and the imbalance is driving the whirl. Figure 2.20a is the plot of the orbit including the transients. Figure 2.20b is the resulting orbit when the initial condition state vector obtained with the convergence algorithm is used with the rotor simulation. Again, Figure 2.20c is the plot of the PSD. Given below in Table 2.5 are the initial condition state vectors for the cases whose plots are presented along with the number of iterations needed for the algorithm to converge to these states.

TABLE 2.5
INITIAL CONDITION STATE VECTORS FOR THE EXAMPLES CITED

RUN	r	$\dot{r}$	ϕ	$\dot{\phi}$	NO.OF ITERATIONS
521	1.8299695 E-3	0.2991794	-2.263848	-271.5819	31
524	2.3821953 E-3	-0.5019839	0.8767700	-275.0212	13
511	3.3850251 E-3	1.564779	-1.098406	1508.773	6
C6	2.3077310 E-4	3.2498761 E-7	5.165721	1256.725	3

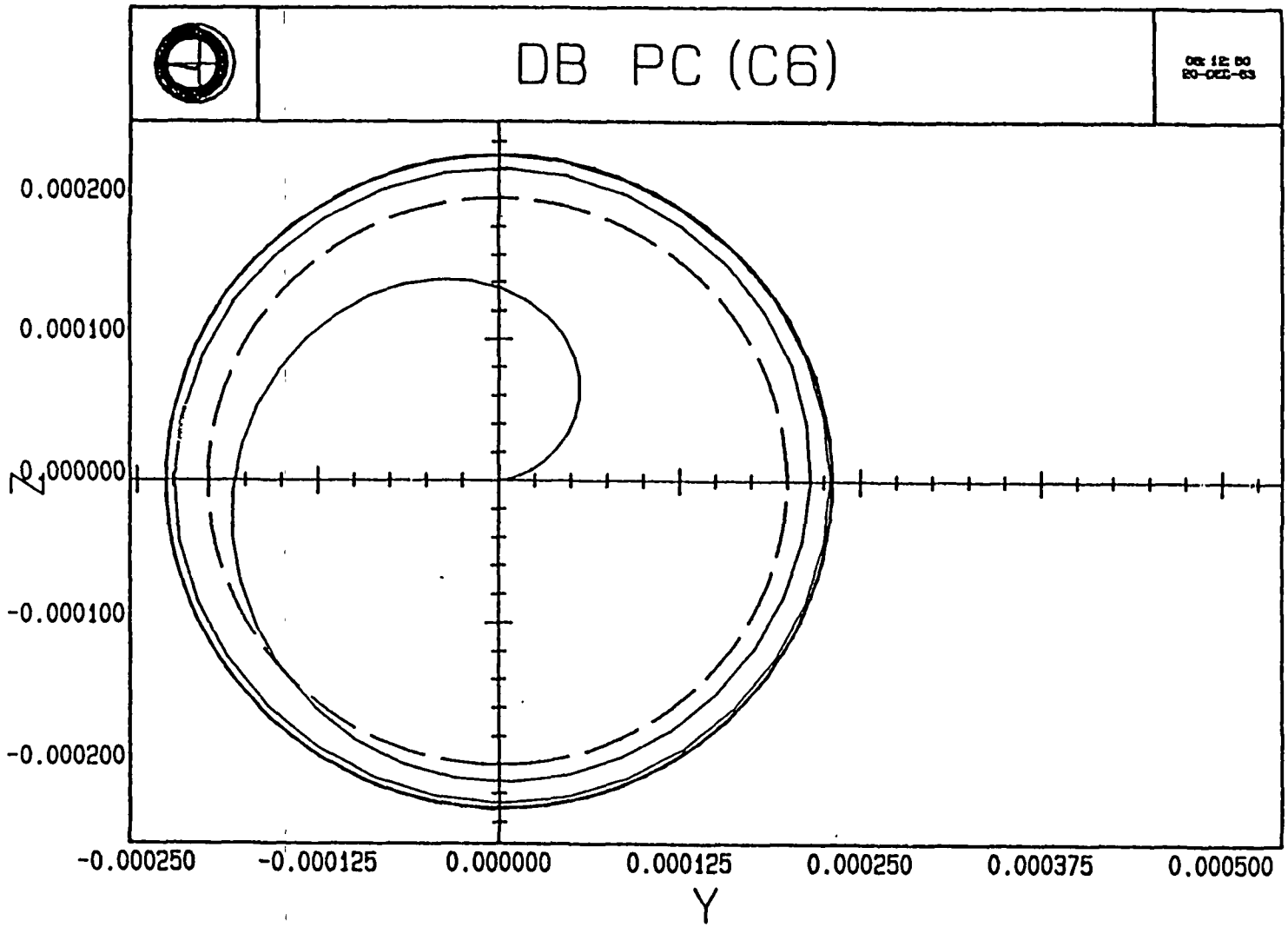

Figure 2.20a Synchronous C-Type Limit Cycle Including Transients.

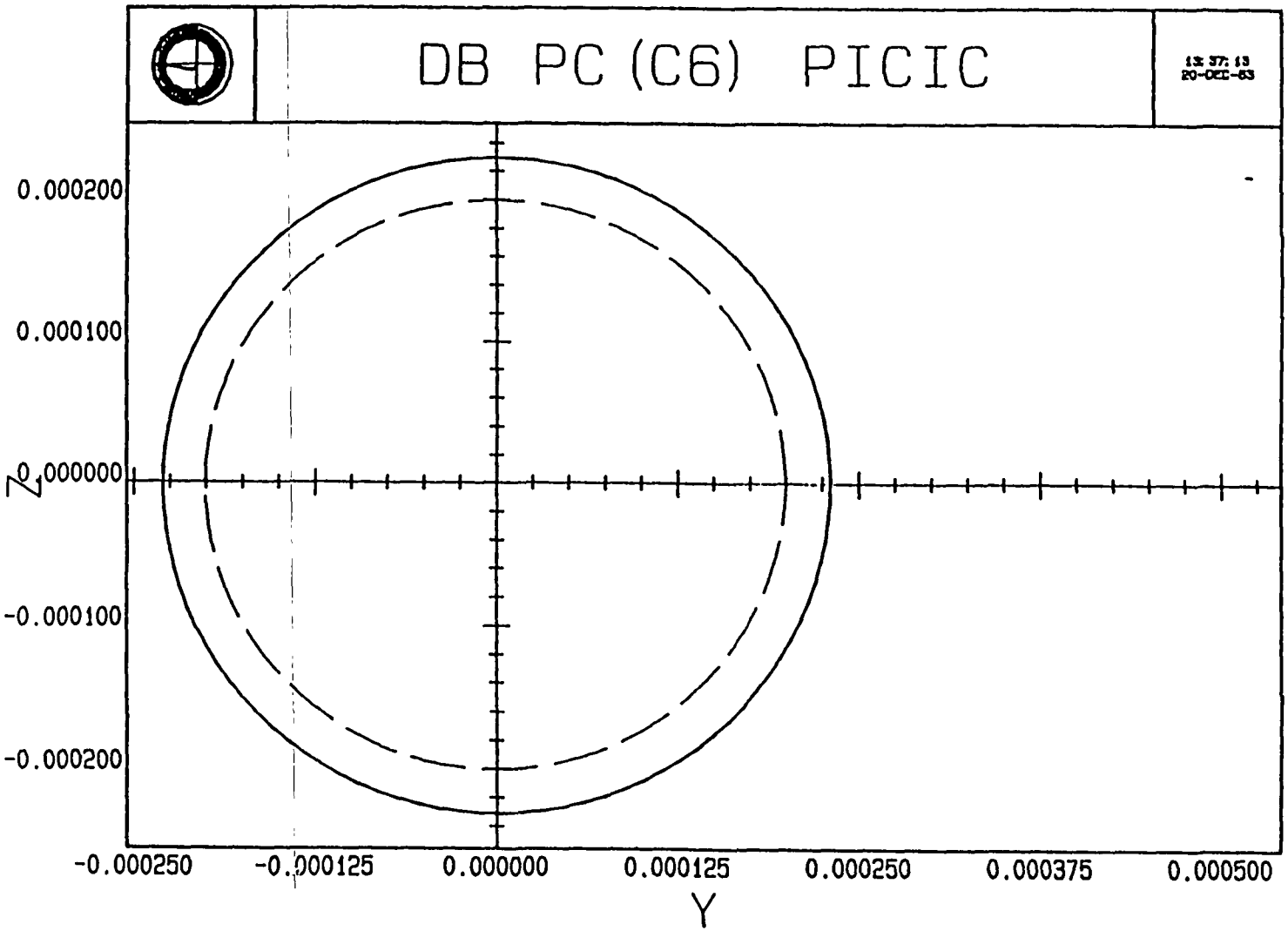

Figure 2.20b Synchronous C-Type Limit Cycle from Convergence Algorithm Initial Conditions.

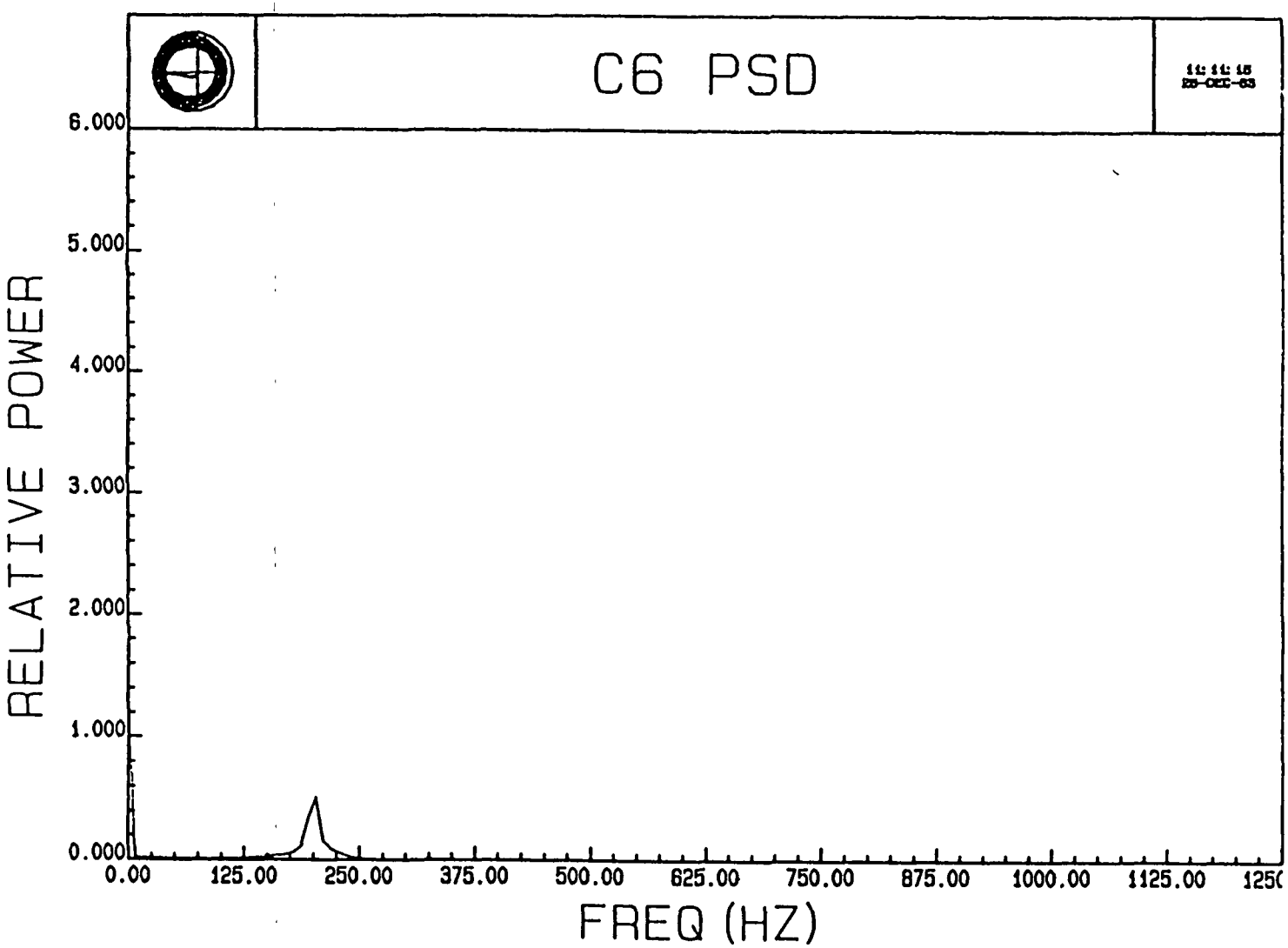

Figure 2.20c PSD for Orbit in 2.20b; Shaft Speed: 200 Hz.

3.0 STABILITY ANALYSIS

An in-depth investigation into the stability properties of the simple rotor system has been conducted. The stability analysis is performed in a stepwise progression, beginning with a study of the system in its simplest form and gradually increasing its complexity until all the parameters are included. Of particular interest is the frequency of instability. Therefore, stability boundaries have been established with respect to frequency as the system parameters are added and varied.

3.1 SIMPLE CASE

The simplest form of the rotor model from which meaningful information may be extracted from its analysis is that in which the deadband, rotor eccentricity, and side forces are all set equal to zero. From this analysis is obtained the global stability boundary, a concept which is further developed in later sections of this document. The equations of motion, in polar coordinates, for the a rotor model are

$$\ddot{r} = \frac{-K}{m} r - \frac{C_S}{m} \dot{r} - \frac{C_Q}{m} r\dot{\phi} + r\dot{\phi}^2 \tag{3.1}$$

$$\ddot{\phi} = \frac{Q_S}{m} - \frac{C_S}{m} \dot{\phi} + \frac{C_Q}{m} \frac{\dot{r}}{r} - \frac{2\dot{r}\dot{\phi}}{r} \tag{3.2}$$

with the seal and bearing stiffnesses are lumped into K. The system described is nonlinear, and, therefore must be linearized in order to determine stability. Small perterbation analysis is employed to perform the linearization. The following definitions are made:

$$\begin{aligned} r &= \delta r + r_o & \dot{\phi} &= \delta\dot{\phi} + \dot{\phi}_o \\ \dot{r} &= \delta\dot{r} & \ddot{\phi} &= \delta\ddot{\phi} \\ \ddot{r} &= \delta\ddot{r} \end{aligned} \tag{3.3}$$

where r_o and $\dot{\phi}_o$ define an equilibrium point and the δ quantities are the perterbation terms. The perterbation equations are

$$\delta\ddot{r} = \frac{-K}{m}(\delta r + r_o) - \frac{C_S}{m}\delta\dot{r} - \frac{C_Q}{m}(\delta r + r_o)(\delta\dot{\phi} + \dot{\phi}_o) + (\delta r + r_o)(\delta\dot{\phi} + \dot{\phi}_o)^2 \tag{3.4}$$

$$\delta\ddot{\phi} = \frac{Q_S}{m} - \frac{C_S}{m}(\delta\dot{\phi} + \dot{\phi}_o) + \frac{C_Q}{m}\frac{\delta r}{(\delta r + r_o)} - \frac{2\,\delta\dot{r}\,(\delta\dot{\phi} + \dot{\phi}_o)}{(\delta r + r_o)} \tag{3.5}$$

By setting the perterbation terms to zero, the equilibrium points of the system are determined. An equilibrium point is

$$r_o = 0 \qquad \text{and} \qquad \dot{\phi}_o = \frac{Q_S}{C_S} \; . \tag{3.6}$$

Equations 3.4 and 3.5 are expanded about using the above equilibrium point. Disregarding all terms of order two or greater, the expressions describing the system become

$$\delta\ddot{r} = \left(\dot{\phi}_o^2 - \frac{K}{m} - \frac{C_Q}{m} \dot{\phi}_o \right) \delta r - \frac{C_s}{m} \delta\dot{r} \tag{3.7}$$

and

$$0 = \left(\frac{C_Q}{m} - 2 \dot{\phi}_o \right) \delta\dot{r}. \tag{3.8}$$

The characteristic equation is obtained using the Laplace transform and Equation 3.7. The equation is

$$s^2 + \frac{C_s}{m} s + \frac{K}{m} + \frac{C_Q}{m} \dot{\phi}_o - \dot{\phi}_o^2 = 0 \tag{3.9}$$

The stability boundary is easily determined by simply finding the first frequency at which a positive root to Equation 3.9 is obtained. Using the parameters given in Section 2.2, the frequency of instability is 4848.10 radians per second, or 771.6 Hertz.

The fluid angular velocity is approximately one-half that of the shaft. For the turbopump seals, the following relationship exists:

$$\dot{\phi}_o = \frac{Q_s}{C_s} \cong \frac{\omega}{2}. \tag{3.10}$$

The natural frequency of the system is determined, assuming the shaft speed is zero, from Equation 3.9 to be

$$\omega_o = \frac{K}{m} = 2424.05 \tag{3.11}$$

for the values of K and m under consideration. Note that this is one half the value of the instability frequency, as is expected. Results obtained with the rotor simulation are in agreement with the results presented above. The plots Figure 3.1 are indicative of the orbits obtained for frequencies below 4848.10 radians per second with Figure 3.2 depicting the general result for frequencies above the stability boundary. Plotted are the whirl orbit, the x and y components of the orbit with time, the magnitude of the shaft displacement vector, r, as a function of time, and the PSD of the x and y components along with that of r, in the a, b, c and d parts of the figures, respectively.

3.2 NONLINEAR CASE - DEADBAND

The next logical step in the stability study is to analyze the effect of adding a deadband to the system. The general response of the simple system with a

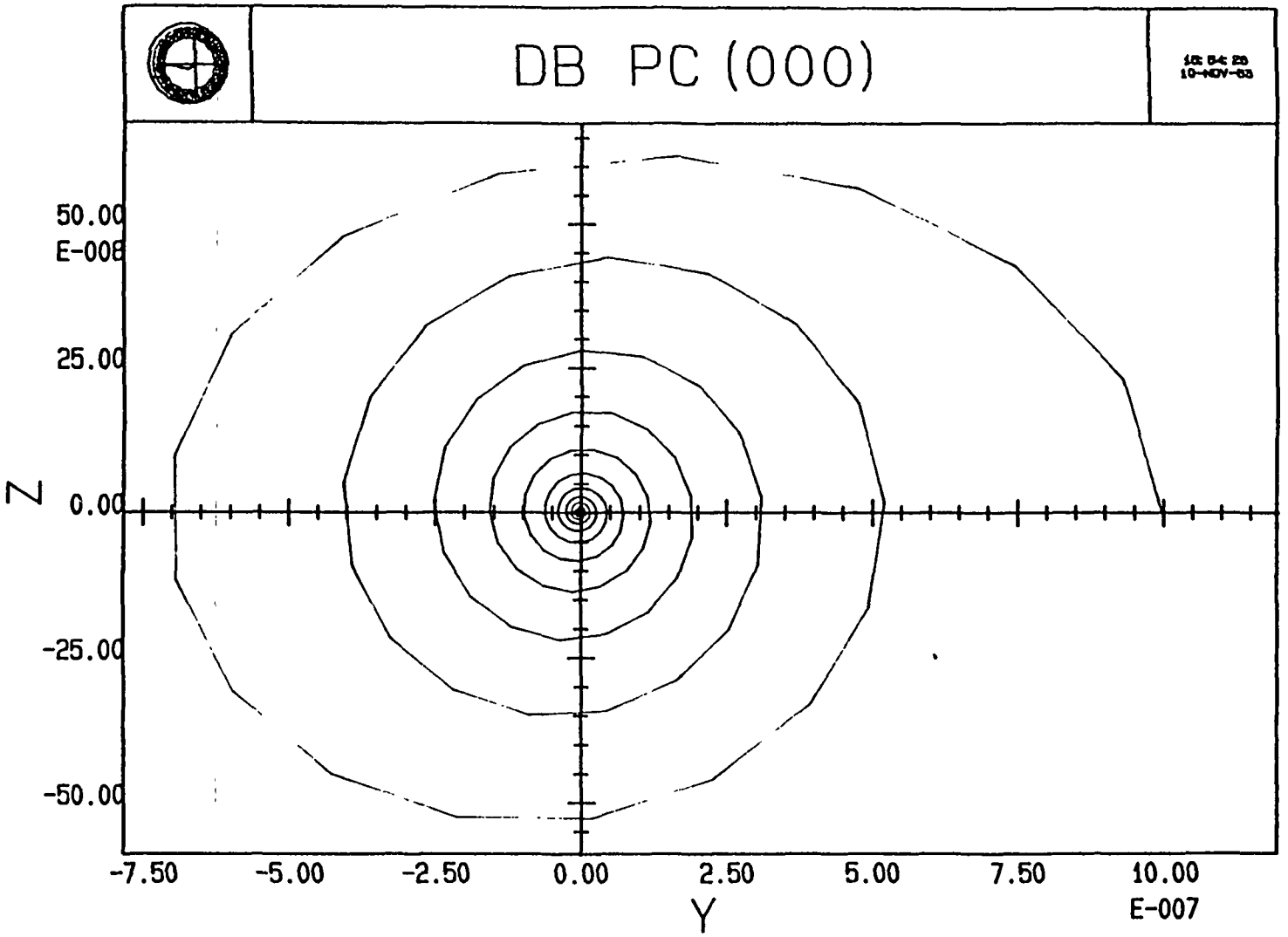

Figure 3.1a Stable Orbit for the Simple System.

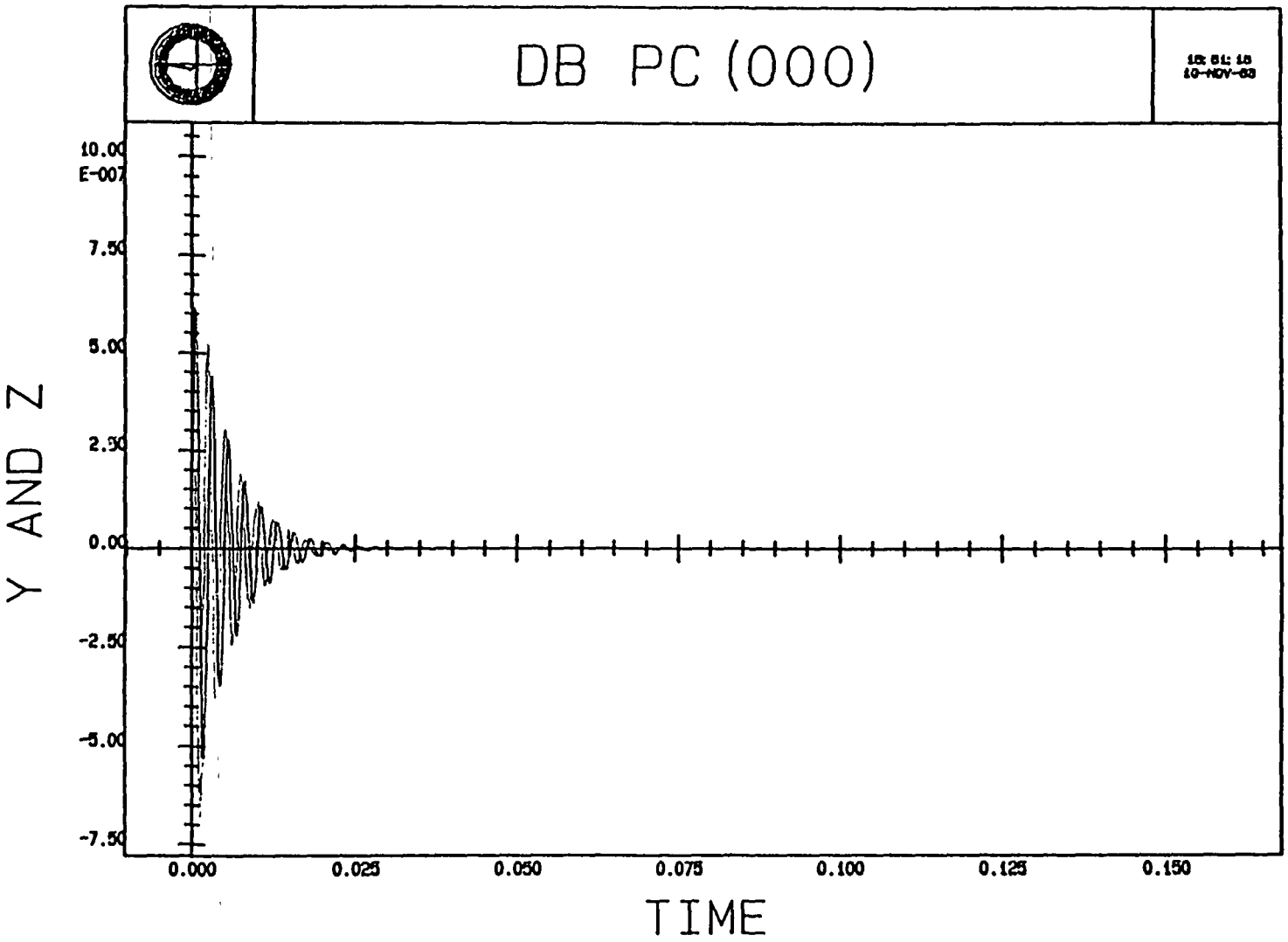

Figure 3.1b The x and y Components as a Function of Time.

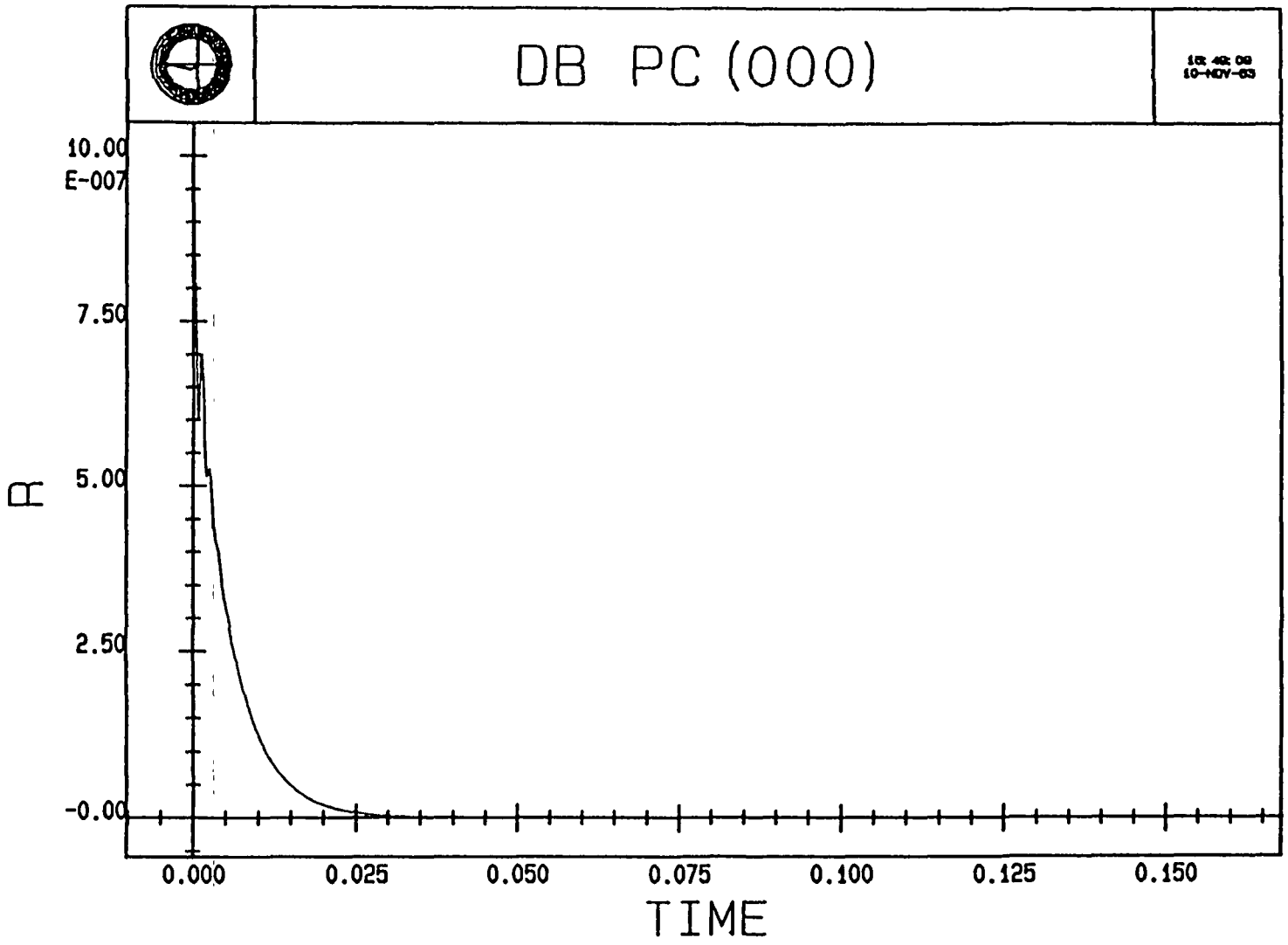

Figure 3.1c The Orbit Radius as a Function of Time.

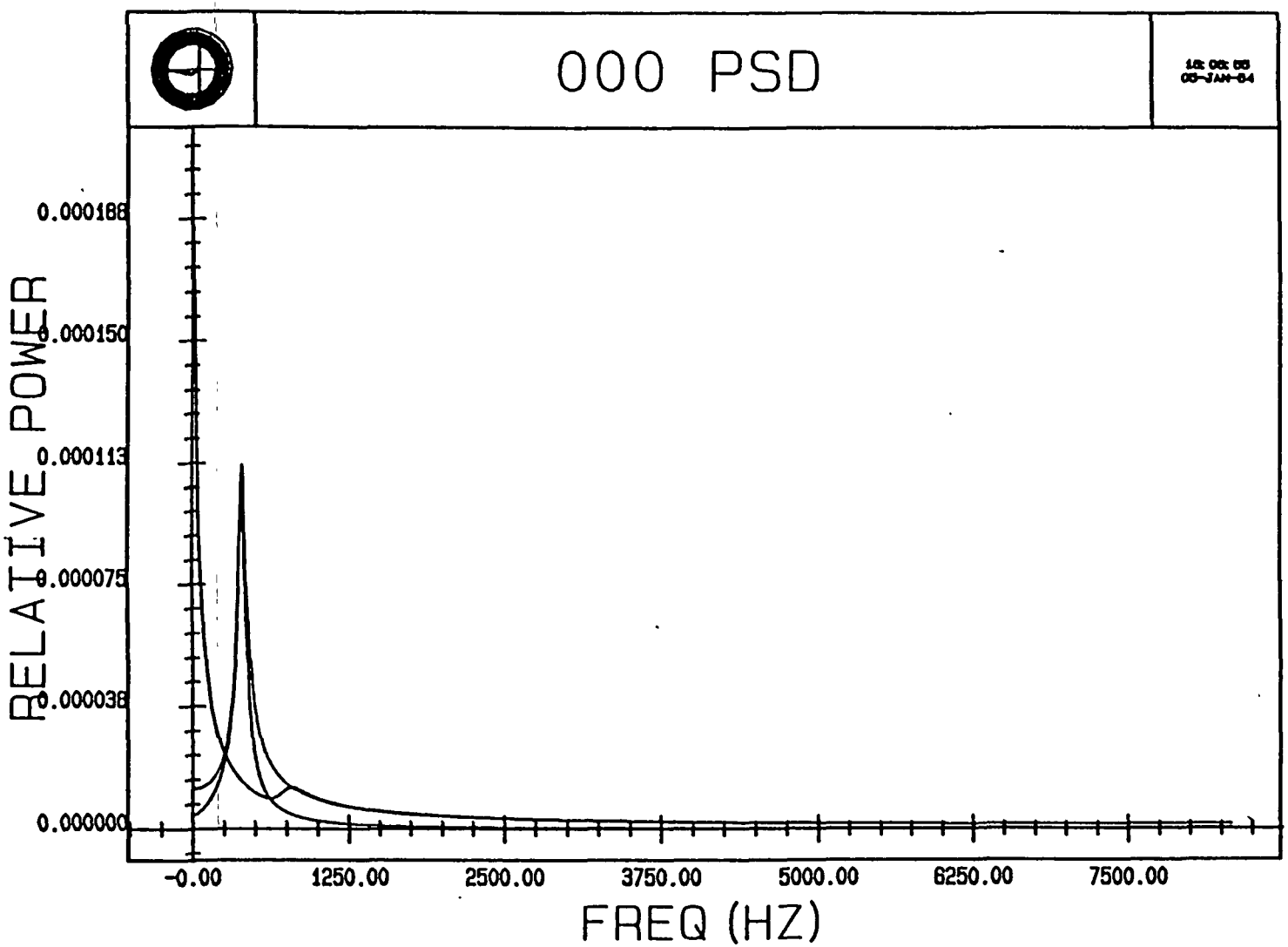

Figure 3.1d PSD of the Stable Orbit; Shaft Speed: 500 Hz.

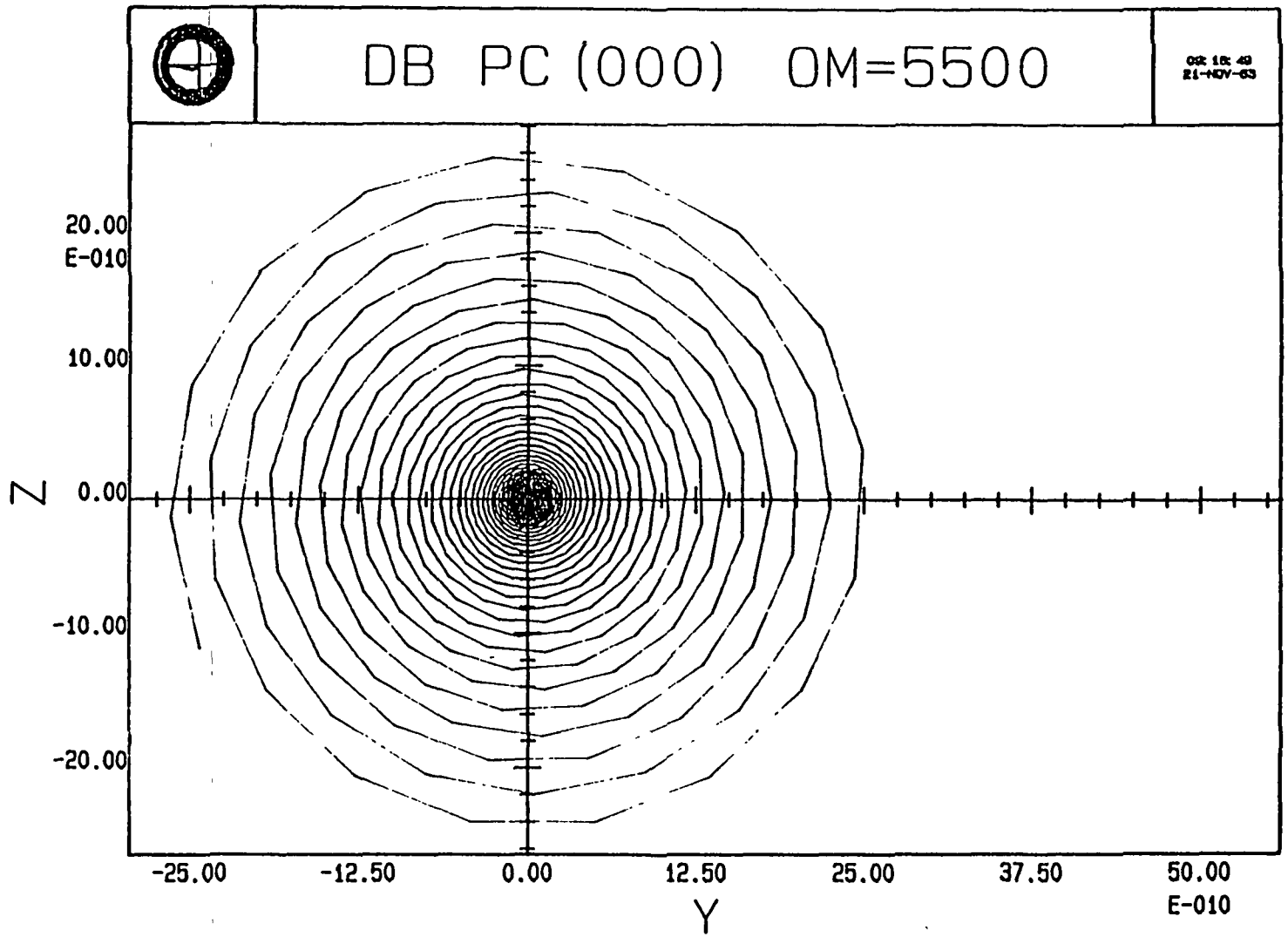

Figure 3.2a Unstable Orbit for the Simple System.

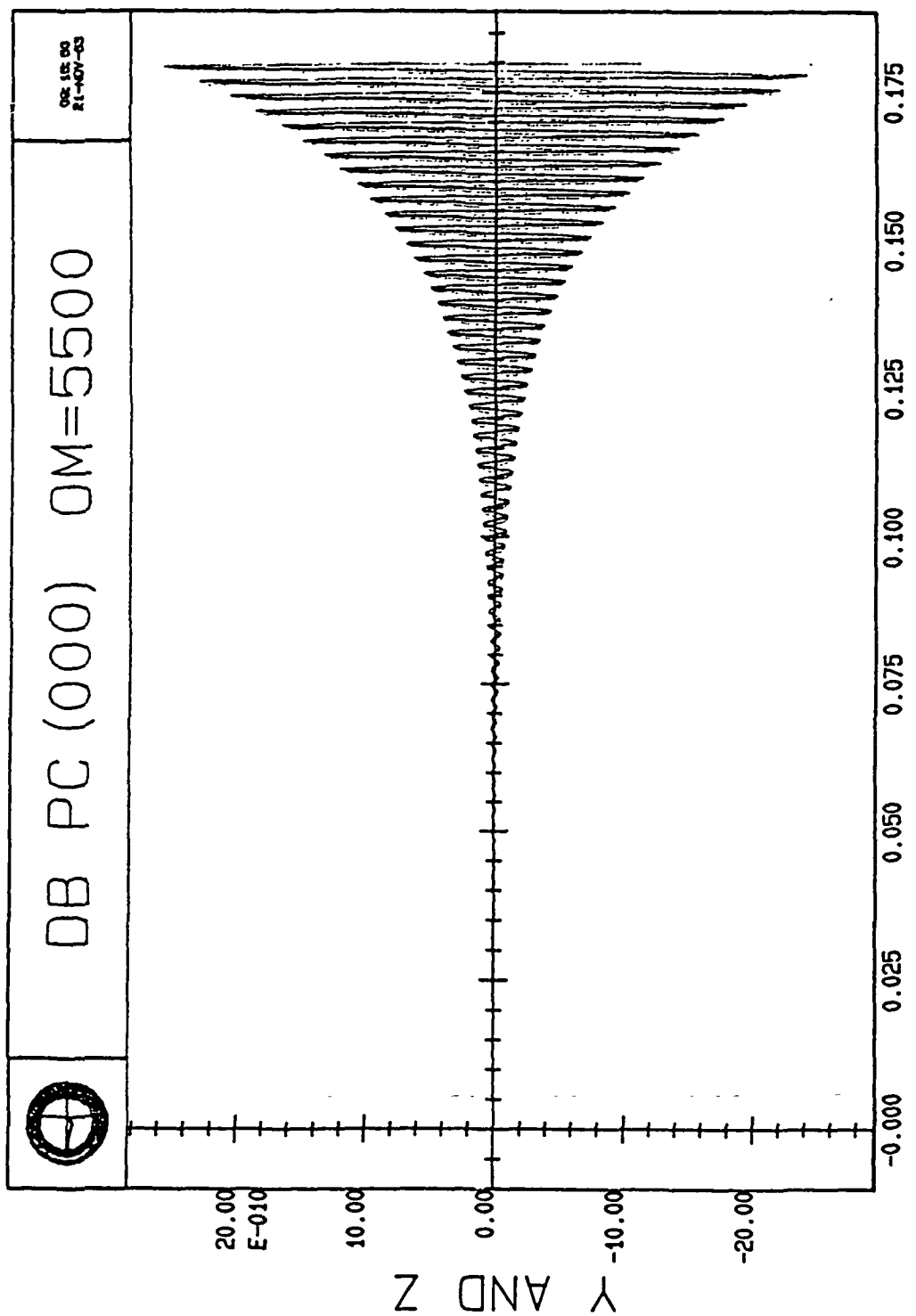

Figure 3.2b The x and y Components as a Function of Time.

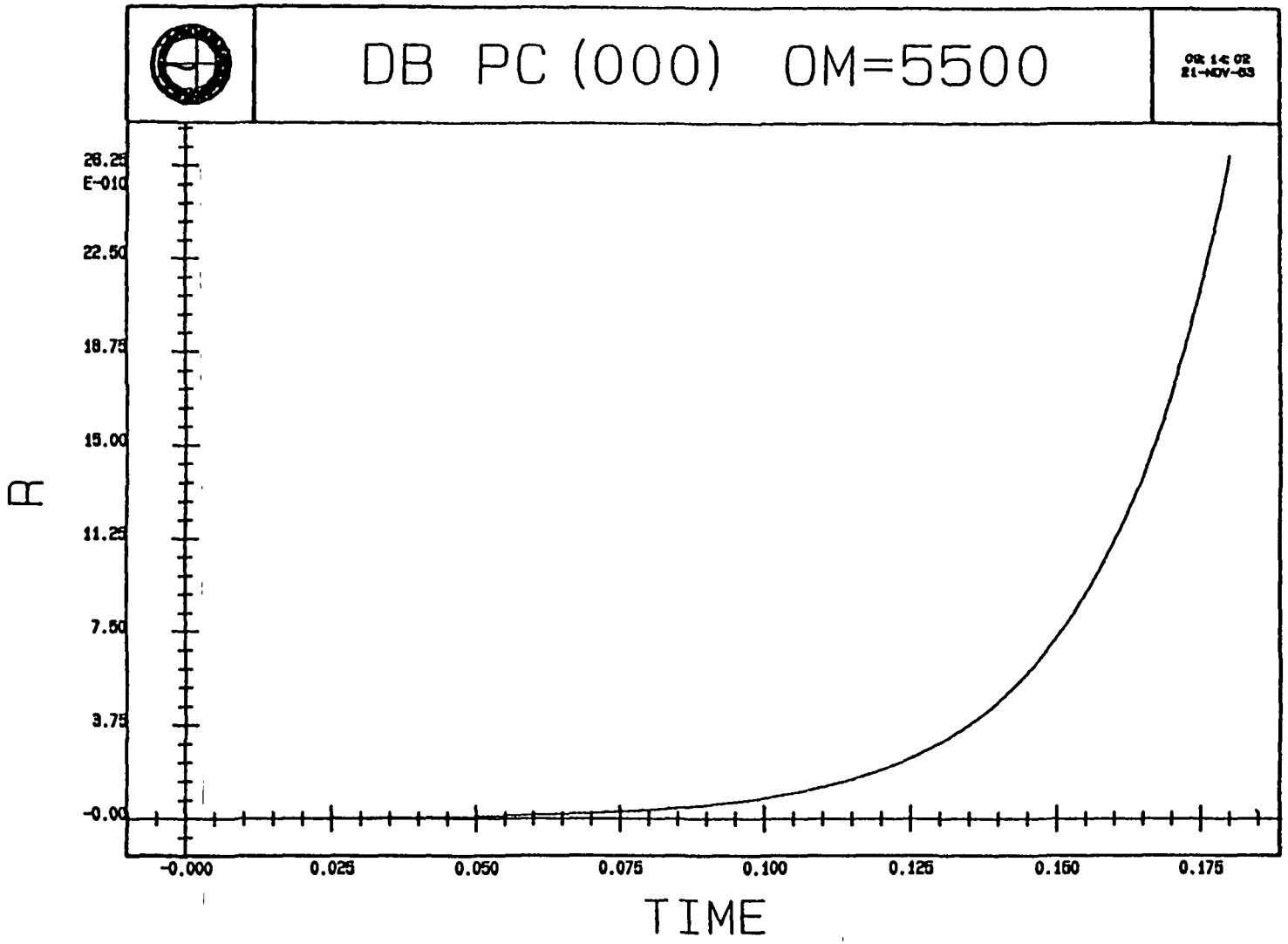

Figure 3.2c The Orbit Radius as a Function of Time.

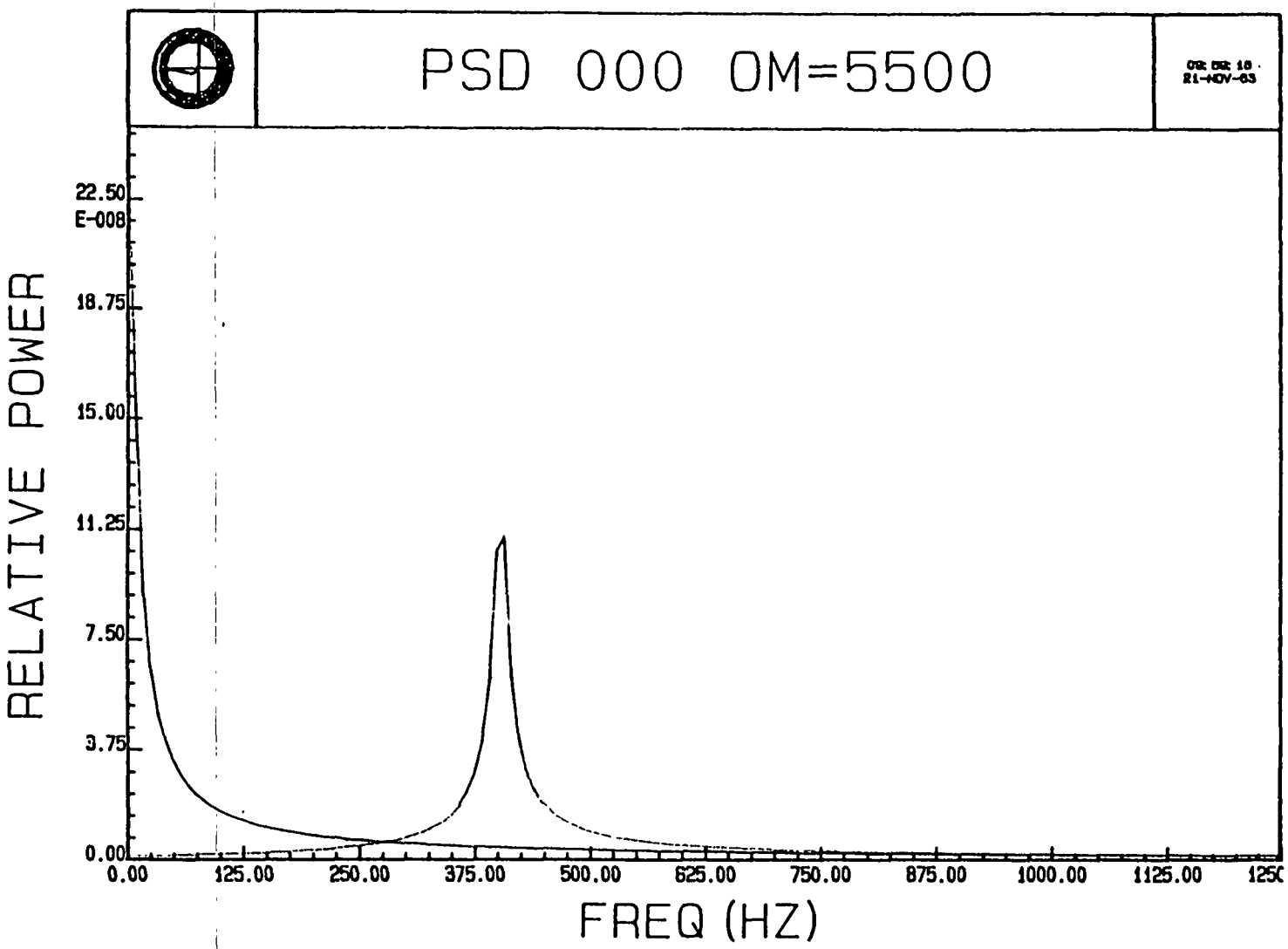

Figure 3.2d PSD of the Stable Orbit; Shaft Speed: 500 Hz.

deadband present is to whirl with a C-type motion at sub-synchronous speed with the whirl radius remaining constant. The differential equations describing the system are, in polar coordinates,

$$\ddot{r} = \frac{-K_B}{m}(r-g)\,u(r-g) - \frac{K_S}{m}\,r - \frac{C_S}{m}\,\dot{r} - \frac{C_Q}{m}\,r\,\dot{\phi} + r\dot{\phi}^2 \tag{3.12}$$

$$\ddot{\phi} = \frac{Q_S}{m} - \frac{C_S}{m}\,\dot{\phi} + \frac{C_Q\,\dot{r}}{m\,r} - \frac{2\dot{r}\dot{\phi}}{r} \tag{3.13}$$

where

$$u(r-g) = \begin{cases} 1 \text{ for } r > g \\ 0 \text{ for } r \leq g \end{cases} \tag{3.14}$$

The equilibrium whirl orbit radius, r_o and the whirl angular rate $\dot{\phi}_o$ are now determined. In order for the system to be in equilibrium, $\dot{r}$ is equal to zero and $\dot{\phi}$ is constant. Imposing these conditions on the above equations, we are able to determine r_o and $\dot{\phi}_o$. First, the value of $\dot{\phi}_o$ is determined from Equation 3.13, to be

$$0 = \frac{Q_S}{m} - \frac{C_S}{m}\,\dot{\phi}_o \tag{3.15}$$

or

$$\dot{\phi}_o = \frac{Q_S}{C_S}\,. \tag{3.16}$$

Using expression 3.16 with Equation 3.12, r_o is determined:

$$0 = \frac{-(K_B + K_S)}{m}\,r + \frac{K_B}{m}\,g - \frac{C_Q\,Q_S}{m\,C_S}\,r + \frac{Q_S^2}{C_S^2}\,r \tag{3.17}$$

Solving for r, which is r_o, we obtain

$$r_o = \frac{K_B C_S^2 g}{C_S^2\,(K_B + K_S) + C_Q\,Q_S\,C_S - Q_S^2\,m} \tag{3.18}$$

We may now go to the simulation to compare these results with those of the program. Again, the parameter values indicated in Table 2.1 are used for various values of deadband. Keep in mind that no imbalance or side forces are under consideration, only the presence of a deadband in the simple rotor system. A table has been constructed which contains the comparison data for the analytical and numerical solutions to r_o and $\dot{\phi}_o$. The second and third columns in Table 3.1

are the analytical and numerical values acquired, respectively. The remaining two columns are values of $\dot{\phi}_o$ for the five different values of deadband considered. The shaft spin rate for this set of runs is 3141.59 radians/second (500 Hertz).

TABLE 3.1
COMPARISON OF ANALYTICAL AND NUMERICAL RESULTS FOR r_o AND ϕ_o

DEADBAND VALUE	ANALYTICAL r_o (in)	NUMERICAL r_o (in)	ANALYTICAL $\dot{\phi}_o$ (rad/sec)	NUMERICAL $\dot{\phi}_o$ (rad/sec)
0.5×10^{-3}	6.5881×10^{-4}	6.5881×10^{-4}	1570.795	1570.789
1.0×10^{-3}	1.3176×10^{-3}	1.3176×10^{-3}	1570.795	1570.789
1.5×10^{-3}	1.9764×10^{-3}	1.9764×10^{-3}	1570.795	1570.788
2.0×10^{-3}	2.6353×10^{-3}	2.6352×10^{-3}	1570.795	1570.789
2.5×10^{-3}	3.2941×10^{-3}	3.2941×10^{-3}	1570.795	1570.790

Examination of Table 3.1 reveals that there is extremely good agreement between the simulation and the theoretical results with regard to the equilibrium conditions. For the sake of illustration, the plots in Figure 3.3 have been included. The figures are numbered using the same convention as those in Figures 3.1 and 3.2. The majority of the orbital plots to follow in this document will be presented in this format.

With the equilibrium conditions firmly established, the stability properties of the system may now be assessed. Because we know the conditions of equilibrium, a linear form of the system is studied to determine the stability properties about r_o and $\dot{\phi}_o$. Recall that when the rotor model is described in cartesian coordinates, the only non-linearity is the deadband. However, the deadband appears in the expression for r_o and the system stability is to be examined about r_o; therefore, the deadband need not be considered in the equations. It is inherent in the analysis.

Equations 3.19 and 3.20 will be considered as describing the system in equilibrium and are presented below.

$$\ddot{y} = \frac{-(K_B + K_S)}{m} y - \frac{C_S}{m} \dot{y} - \frac{Q_S}{m} z - \frac{C_Q}{m} \dot{z} \qquad (3.19)$$

$$\ddot{z} = \frac{-(K_B + K_S)}{m} z - \frac{C_S}{m} \dot{z} + \frac{Q_S}{m} y + \frac{C_Q}{m} \dot{y} \qquad (3.20)$$

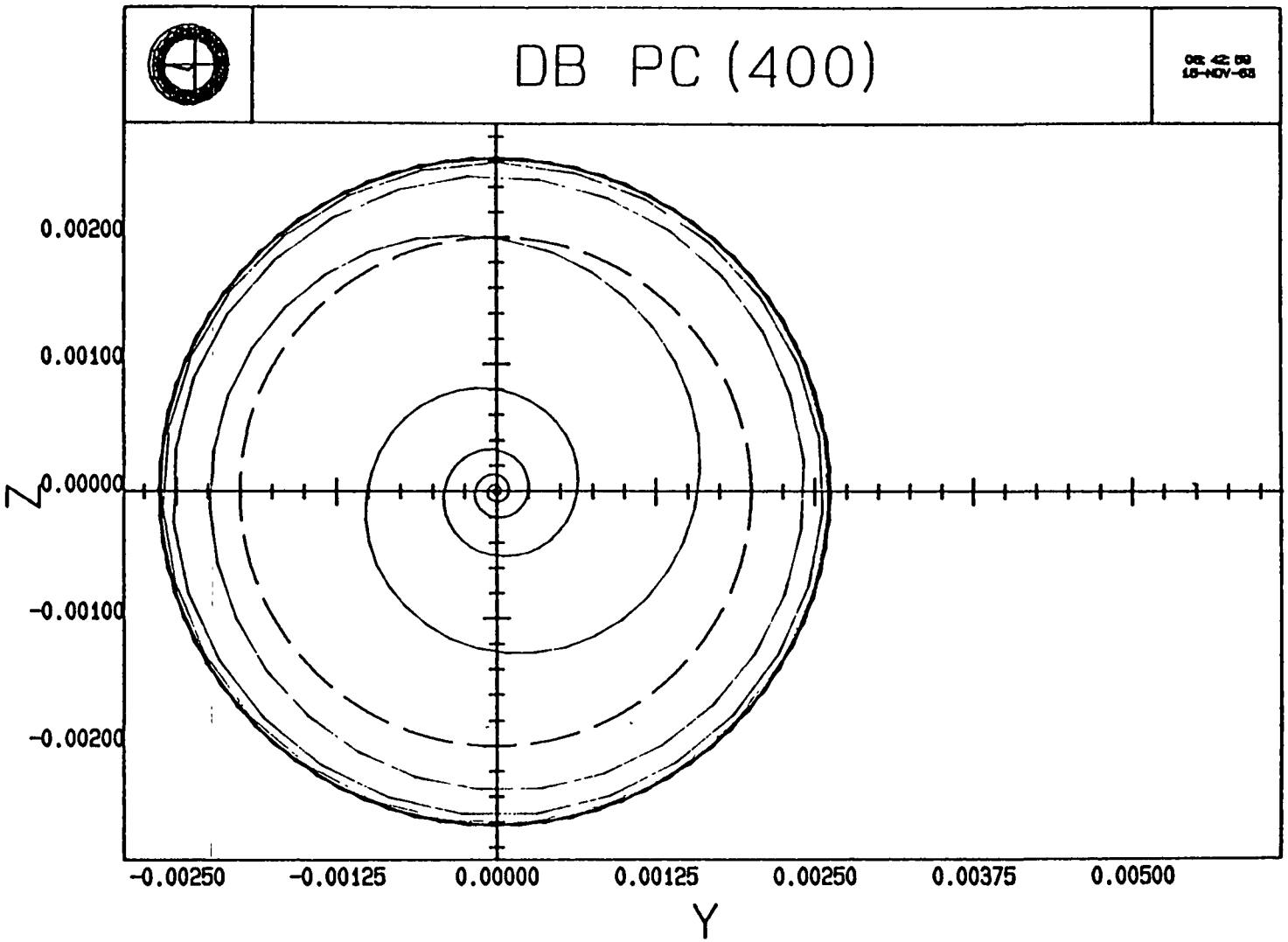

Figure 3.3a Equilibrium Radius for the Simple System Including the Deadband.

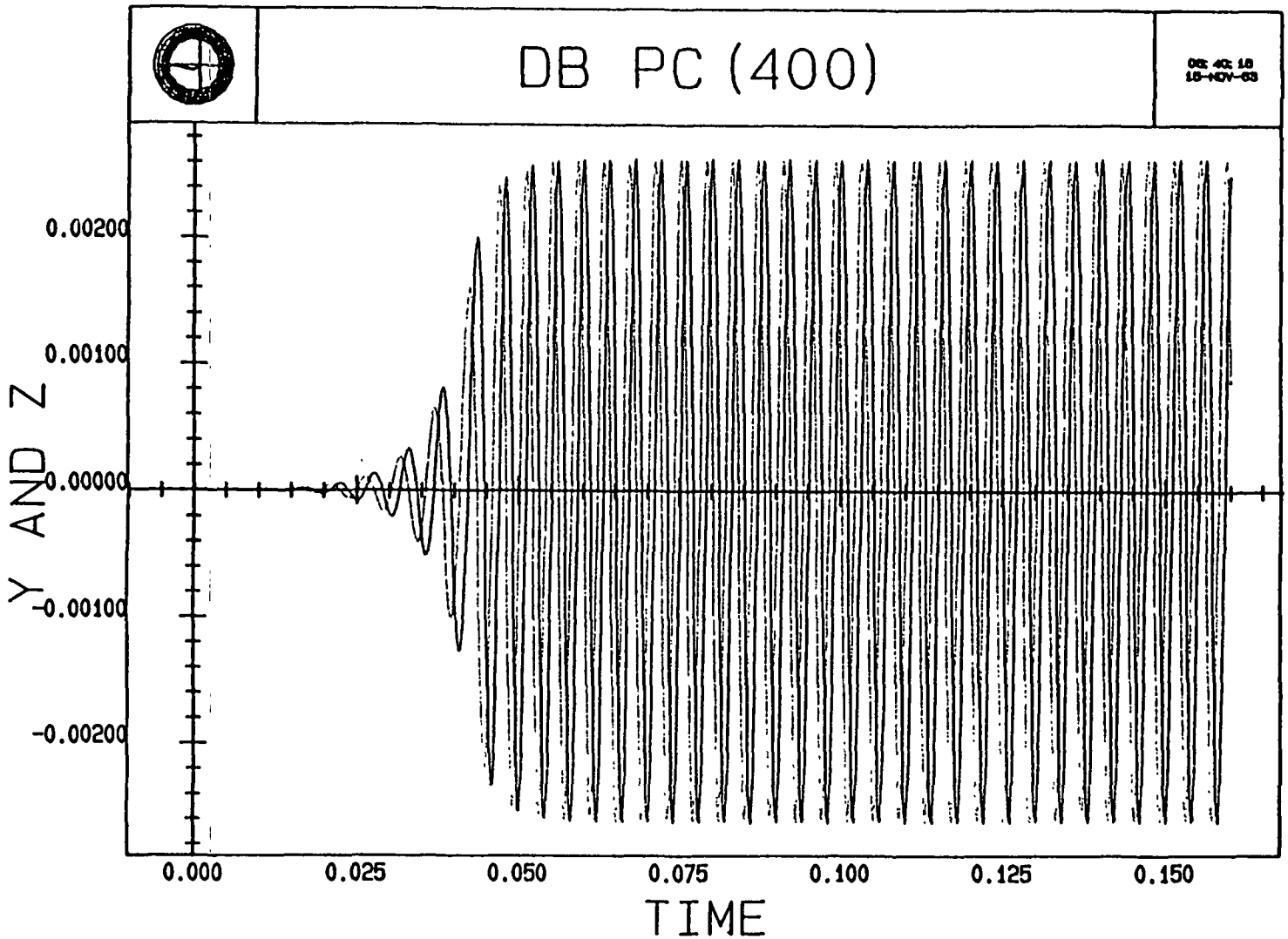

Figure 3.3b The x and y Components as a Function of Time.

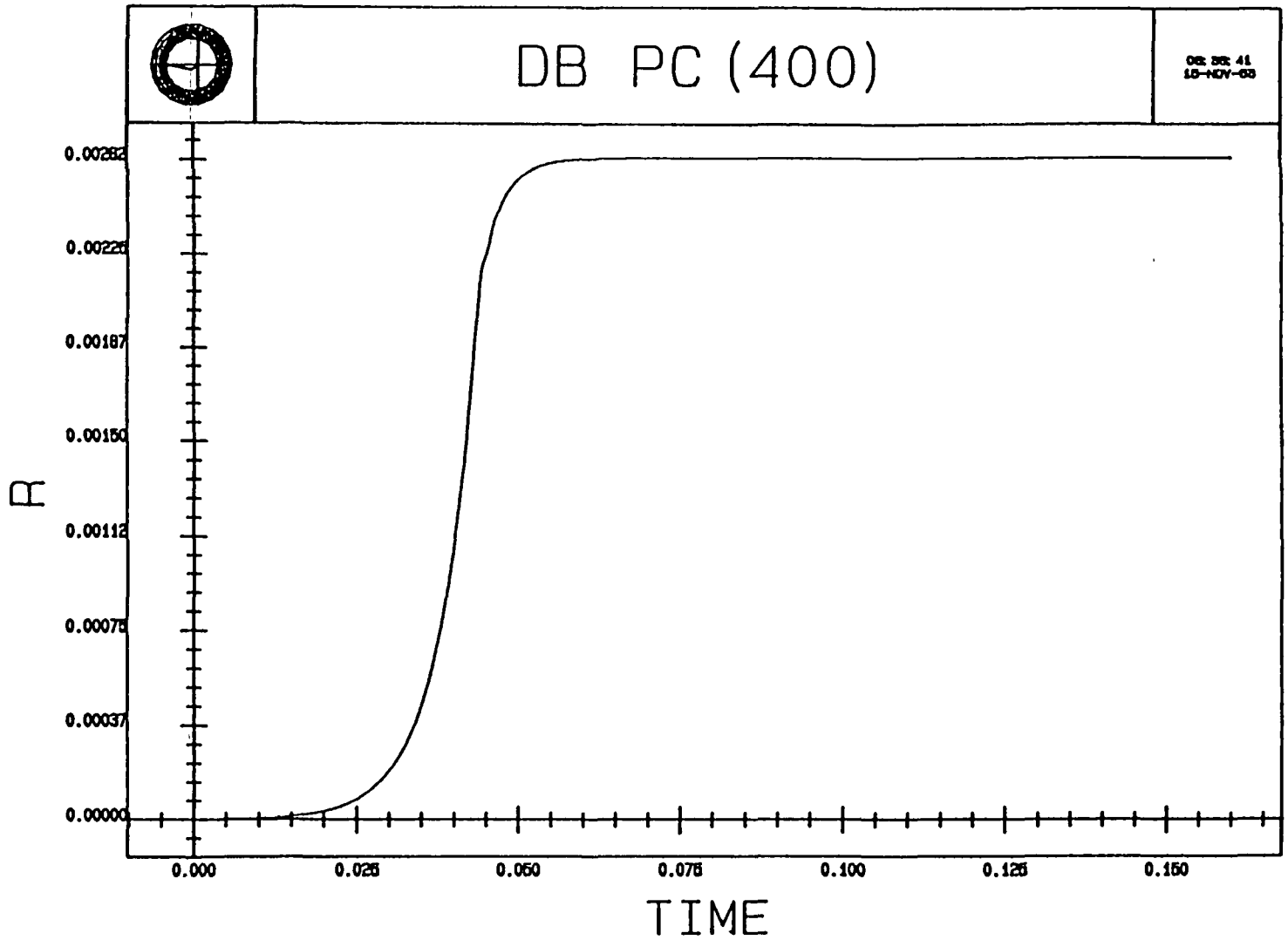

Figure 3.3c The Orbit Radius as a Function of Time.

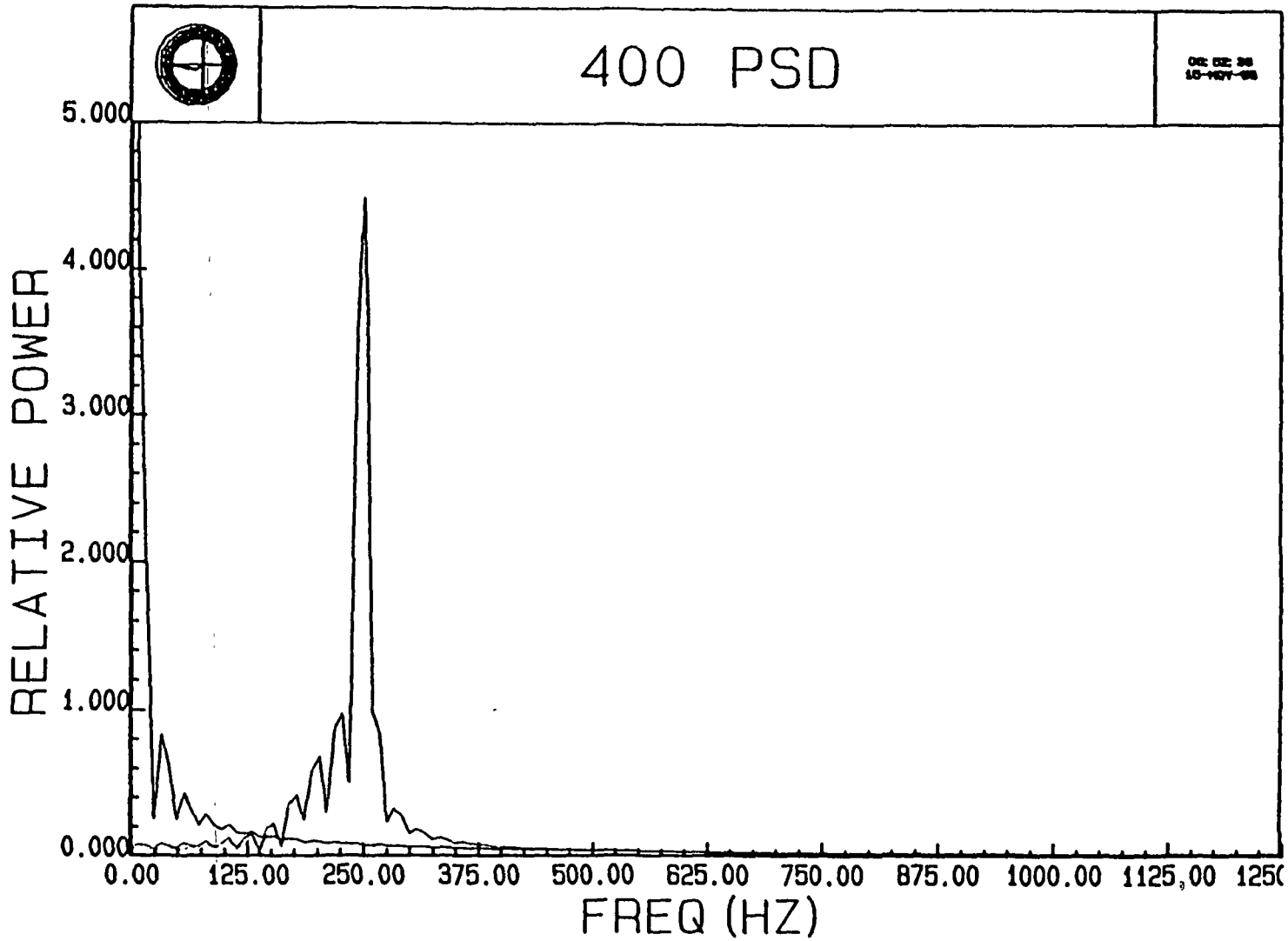

Figure 3.3d PSD of the Equilibrium Orbit; Shaft Speed: 500 Hz.

With the unit vectors $\underline{e}_y$ and $\underline{e}_z$ defined as indicated in Figure 3.4, the following definitions are made for the perterbations.

$$\begin{aligned} y &= \delta r & z &= r_o\,\delta\phi \\ \dot{y} &= \delta\dot{r} & \dot{z} &= \delta\dot{\phi} \\ \ddot{y} &= \delta\ddot{r} & \ddot{z} &= \delta\ddot{\phi} \end{aligned} \tag{3.21}$$

Equations 3.19 and 3.20 become, therefore

$$\delta\ddot{r} = \frac{-(K_B + K_S)}{m}\delta r - \frac{C_S}{m}\delta\dot{r} - \frac{Q_S}{m} r_o\delta\phi - \frac{C_Q}{m}\dot{\delta\phi} \tag{3.22}$$

$$\ddot{\delta\phi} = \frac{-(K_B + K_S)}{m} r_o\,\delta\phi - \frac{C_S}{m}\dot{\delta\phi} + \frac{Q_S}{m}\delta r + \frac{C_Q}{m}\delta\dot{r}\ . \tag{3.23}$$

The approach taken to determine stability is to first cast the two differential equations into a state variable form, with four differential equations resulting. Using state variable analysis techniques, the system is expressed in the form

$$\dot{\underline{x}} = [A]\underline{x} \tag{3.24}$$

where [A] is the system matrix. From such a formulation, the characteristic equation is easily determined by taking the determinant of the matrix [sI - A], where s is the Laplace tranform variable, and [I] is the identity matrix. The following state variables are assigned to the perterbations:

$$\begin{aligned} x_1 &= \delta r & x_3 &= \delta\phi \\ x_2 &= \delta\dot{r} & x_4 &= \delta\dot{\phi} \end{aligned} \tag{3.25}$$

The state variable formulation is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{K}{m} x_1 - \frac{C_S}{m} x_2 - \frac{Q_S r_o}{m} x_3 - \frac{C_Q}{m} x_4 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{Q_S}{m} x_1 + \frac{C_Q}{m} x_2 - \frac{K r_o}{m} x_3 - \frac{C_S}{m} x_4 \end{aligned} \tag{3.26}$$

where $K = K_B + K_S$. The resulting system matrix is

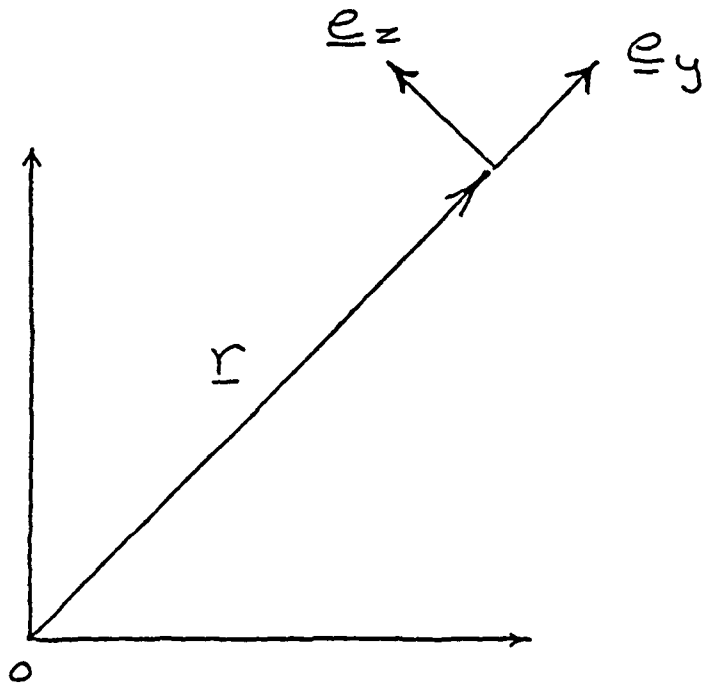

Figure 3.4 Unit Vector Definition.

$$[A] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K}{m} & -\frac{C_S}{m} & -\frac{Q_S r_o}{m} & -\frac{C_Q}{m} \\ 0 & 0 & 0 & 1 \\ \frac{Q_S}{m} & \frac{C_Q}{m} & -\frac{K r_o}{m} & -\frac{C_S}{m} \end{bmatrix} . \qquad (3.27)$$

The characteristic equation, P(s), is, therefore

$$P(s) = \det[sI - A] = \begin{vmatrix} s & -1 & 0 & 0 \\ \frac{K}{m} & s + \frac{C_S}{m} & \frac{Q_S r_o}{m} & \frac{C_Q}{m} \\ 0 & 0 & s & -1 \\ -\frac{Q_S}{m} & -\frac{C_Q}{m} & \frac{K r_o}{m} & s + \frac{C_S}{m} \end{vmatrix} \qquad (3.28)$$

or, performing the indicated operation,

$$P(s) = s^4 + \frac{2C_S}{m} s^3 + \frac{C_S^2 + C_Q^2 + mK(r_o + 1)}{m^2} s^2 + \frac{(Q_S C_Q + C_S K)(r_o + 1)}{m^2} s + \frac{(Q_S^2 + K^2) r_o}{m^2} \qquad (3.29)$$

The stability boundary for the system may now be established via determination of the first frequency at which the above equation has a positive root. Recall that the values of both C_S and Q_S are dependent upon the shaft angular velocity. By varying these two parameters and examining the roots of the above equation, the instability frequency is determined. It turns out that this frequency is not dependent upon r_o and has been determined to be: ω = 5047.93 radians/sec (803.4 Hertz).

Apparently, the addition of the deadband enhances the local stability of the system. That is, the stability of the system with regard to the equilibrium conditions stated. Verification of the stability boundary may be achieved by examination of the resulting orbits obtained when the simulation is executed for shaft speed at or above 5047.93 radians/second. The orbit plotted in figure 3.5 is the result from the simulation executed with a shaft speed of 4800 radians/second (764 Hertz). Program output indicates that an equilibrium radius of 8.355×10^{-3} inches is approached with the value of $\dot{\phi}$ being 2400.0 radians/second. The system is stable. When the simulation is executed at a shaft speed of 5040 (802 Hertz), the system is still stable, approaching an equilibrium radius of 0.23 inches, as indicated by the plot in Figure 3.6. Only the magnitude of the radius vector is shown for this run. Even though the system is stable in the analysis the bearing loads would obviously be intolerable for such a large orbit radius. Instability results at a shaft spin rate of 5047.93 radians/second, as predicted in the analysis. The resulting whirl orbit for this case is plotted in Figure 3.7. The character of orbits generated at frequencies above 5047.93 radians per second is not unlike the one shown in Figure 3.8. The program is executed with a shaft angular velocity of 5500 radians/second (875 Hertz) to produce these results.

It is interesting to note that when the stability analysis is performed for this particular configuration in polar coordinates, the resulting characteristic equation is third order, rather than fourth order. Briefly, we begin with Equations 3.12 and 3.13 and use small perturbation analysis with the variable assignments listed below.

$$\begin{aligned} r &= \delta r + r_o & \phi &= \delta\phi + \dot{\phi}_o t \\ \dot{r} &= \delta\dot{r} & \dot{\phi} &= \delta\dot{\phi} + \dot{\phi}_o \\ \ddot{r} &= \delta\ddot{r} & \ddot{\phi} &= \delta\ddot{\phi} \end{aligned} \tag{3.30}$$

Note, however, that nowhere in equations 3.12 or 3.13 appears the variable ϕ. Therefore the only state assignments necessary are the ones for δr, $\delta\dot{r}$ and $\delta\dot{\phi}$. The equilibrium conditions are obtained by setting the perturbations to zero, with the results identical to the expressions for $\dot{\phi}_o$ and r_o given in equations 3.16 and 3.18, respectively. The perturbation equations are

$$\delta\ddot{r} = -\left[\frac{-(K_B + K_S)}{m} - \frac{C_Q}{m}\dot{\phi}_o + \dot{\phi}_o^2\right]\delta r - \frac{C_S}{m}\delta\dot{r} + \left[2\dot{\phi}_o r_o - \frac{C_Q}{m} r_o\right]\delta\dot{\phi}$$

$$+ \frac{K_B g - (K_B + K_S) r_o - C_Q r_o \dot{\phi}_o}{m} + r_o\dot{\phi}_o^2 \tag{3.31}$$

$$\delta\ddot{\phi} = \frac{Q_S - C_S\dot{\phi}_o}{m r_o}\delta r + \frac{C_Q}{r_o m}\delta\dot{r} - \frac{C_S}{m}\delta\dot{\phi} + \frac{Q_S - C_S\dot{\phi}_o}{m} . \tag{3.32}$$

Let

$$\begin{aligned} x_1 &= \delta r \\ x_2 &= \delta\dot{r} \\ x_3 &= \delta\dot{\phi} \end{aligned} \tag{3.33}$$

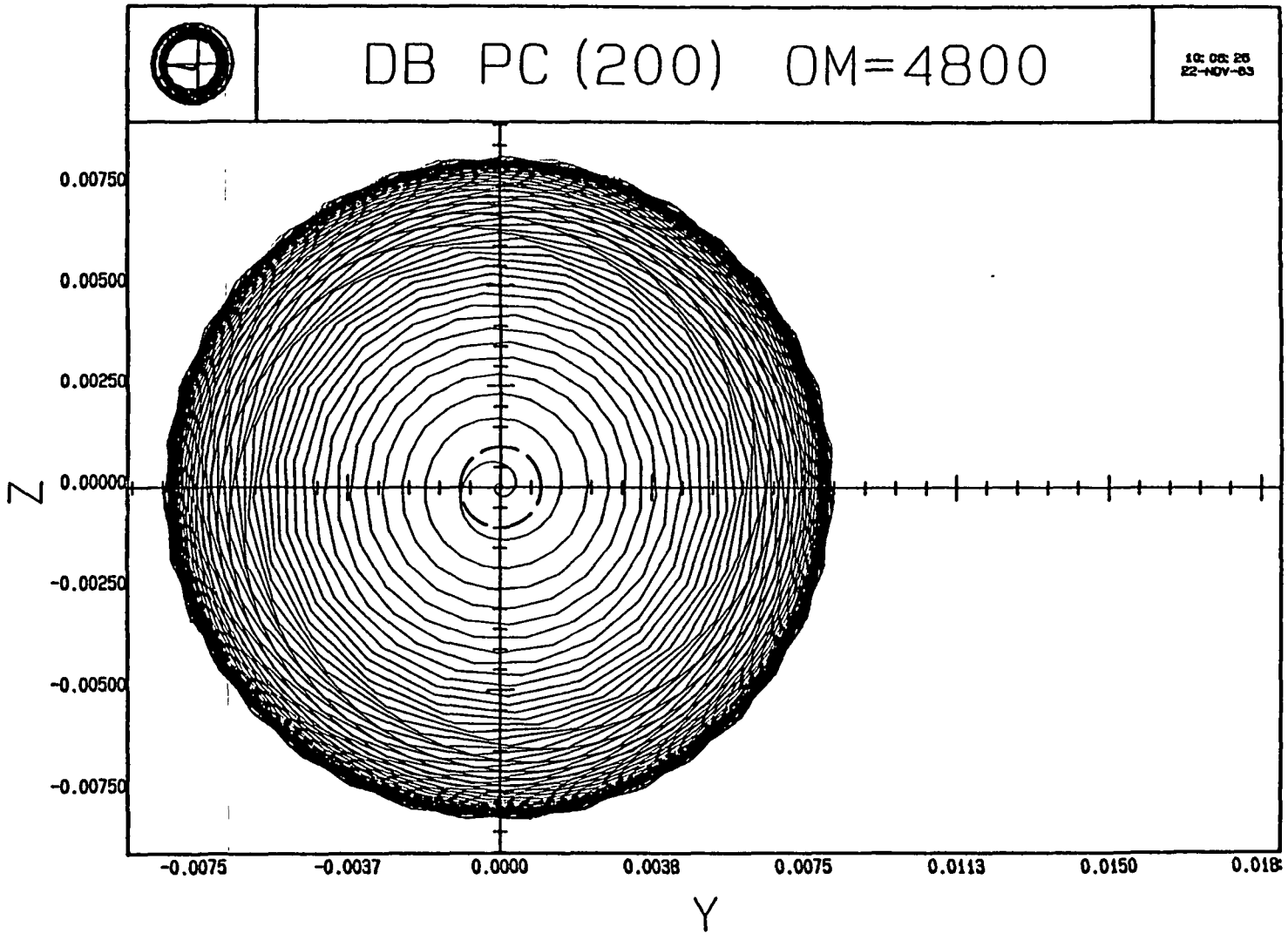

Figure 3.5a Orbit Approaching Equilibrium.

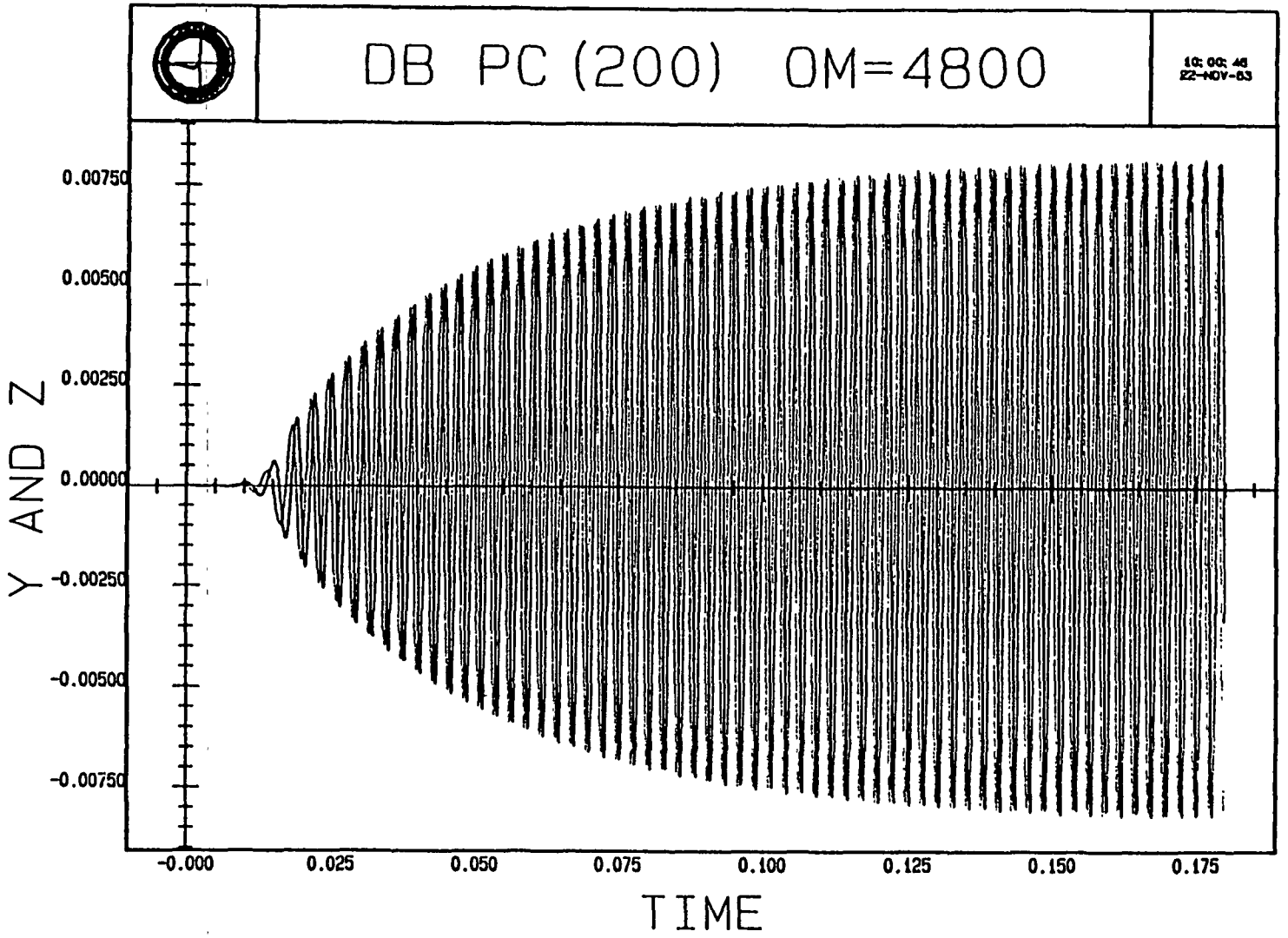

Figure 3.5b The x and y Components as a Function of Time.

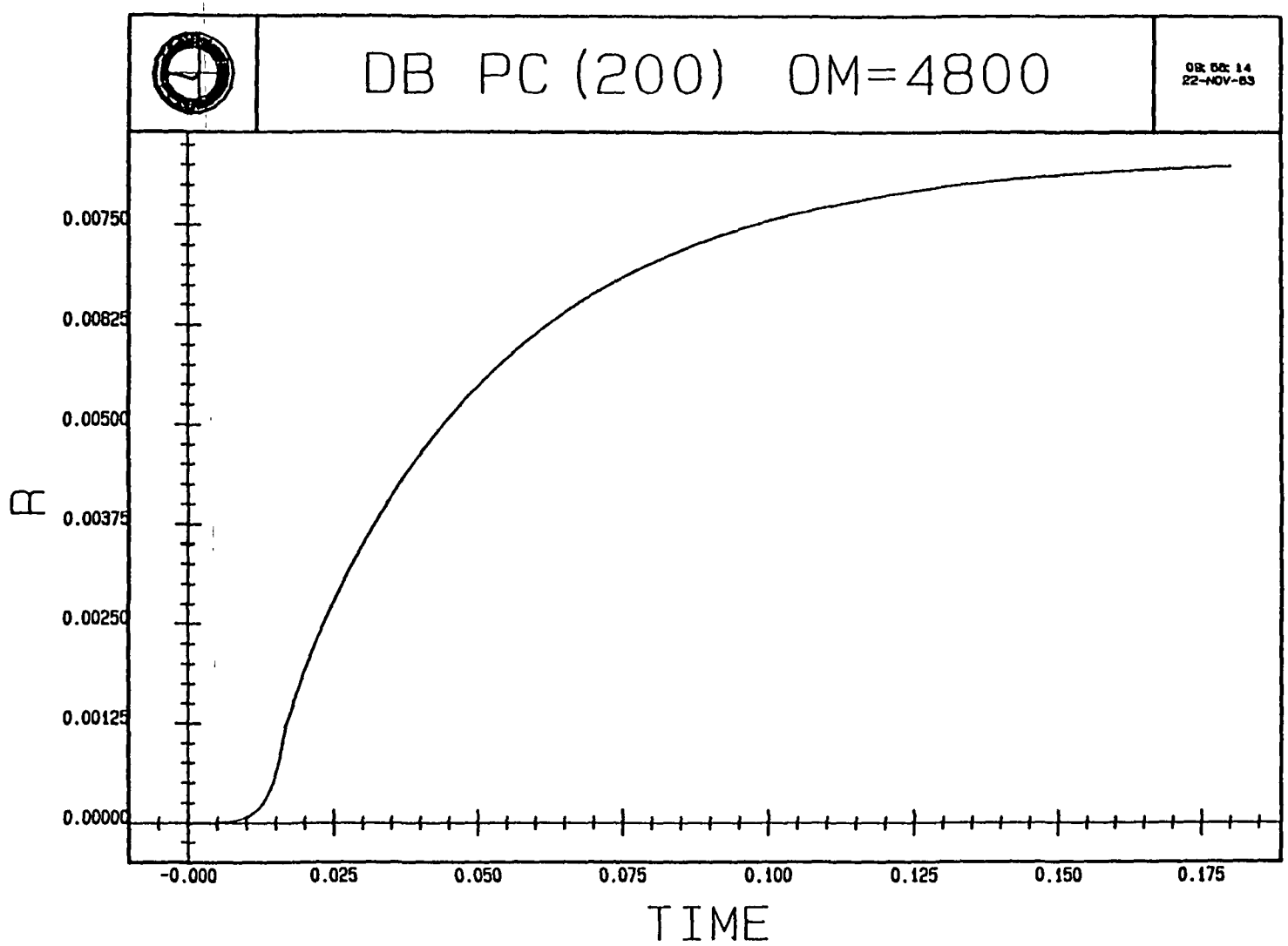

Figure 3.5c The Orbit Radius as a Function of Time.

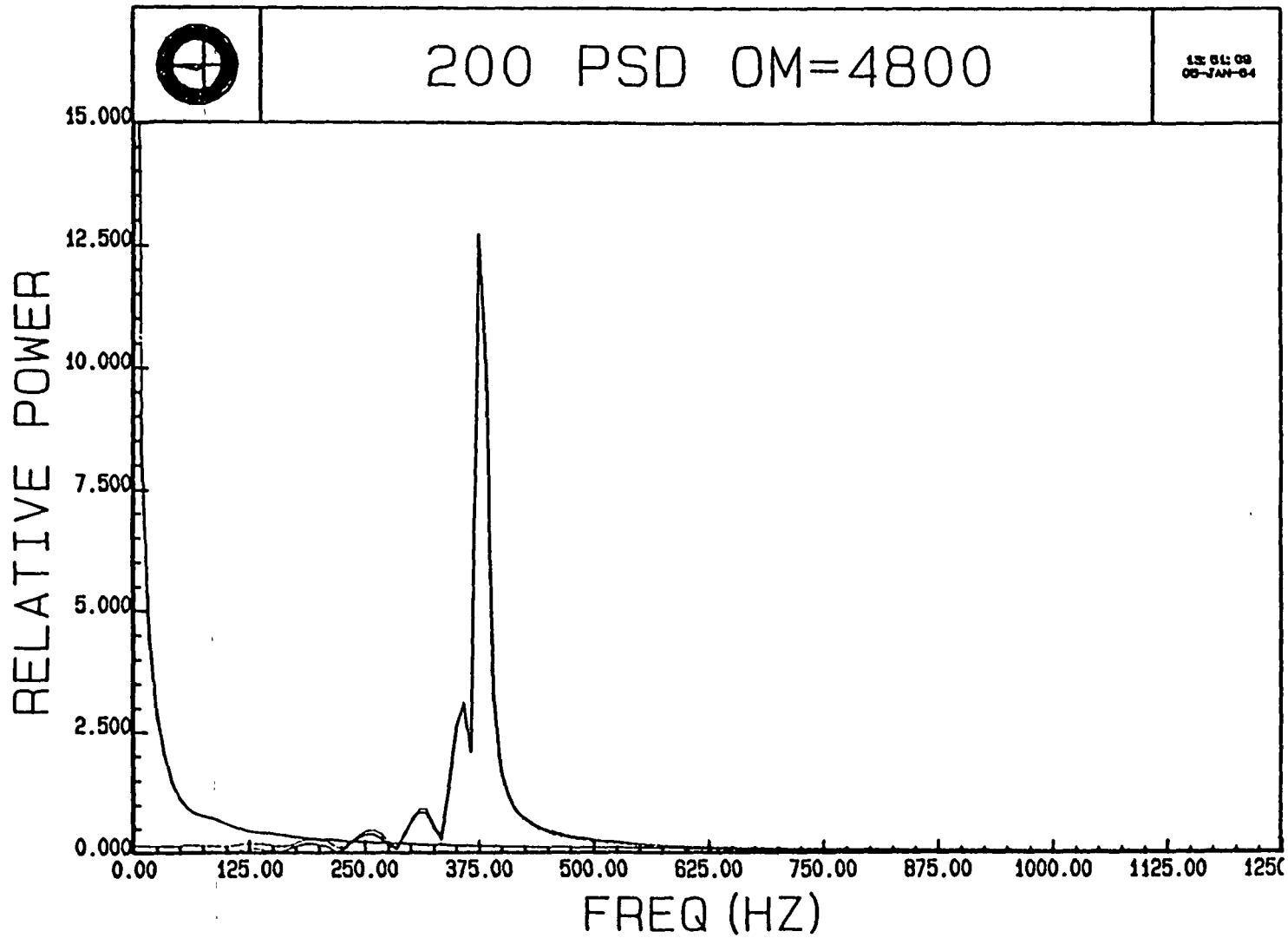

Figure 3.5d PSD of the Equilibrium Orbit; Shaft Speed: 764 Hz.

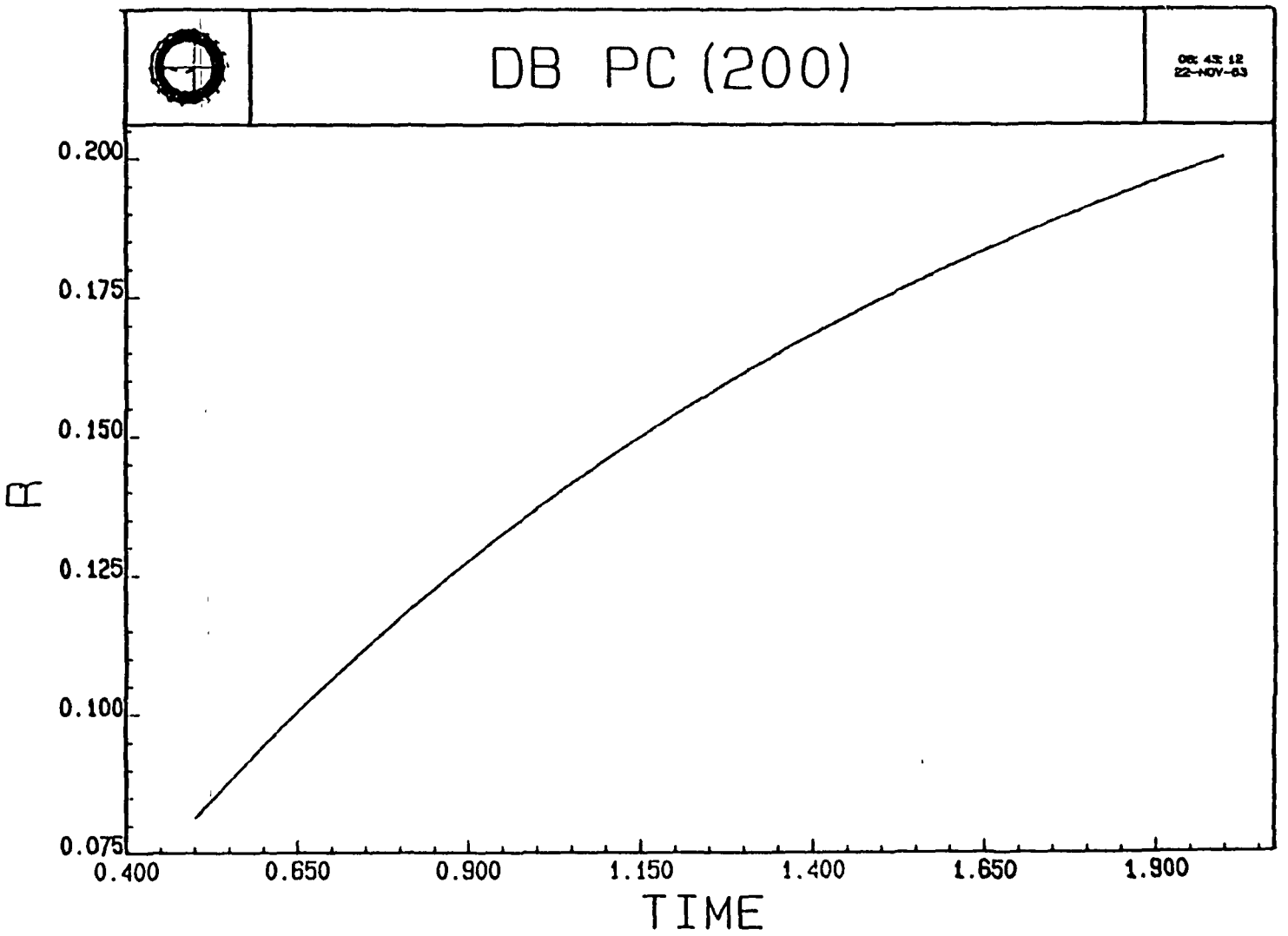

Figure 3.6 Orbital Radius vs. Time; Shaft Speed: 5040 Radians/Second.

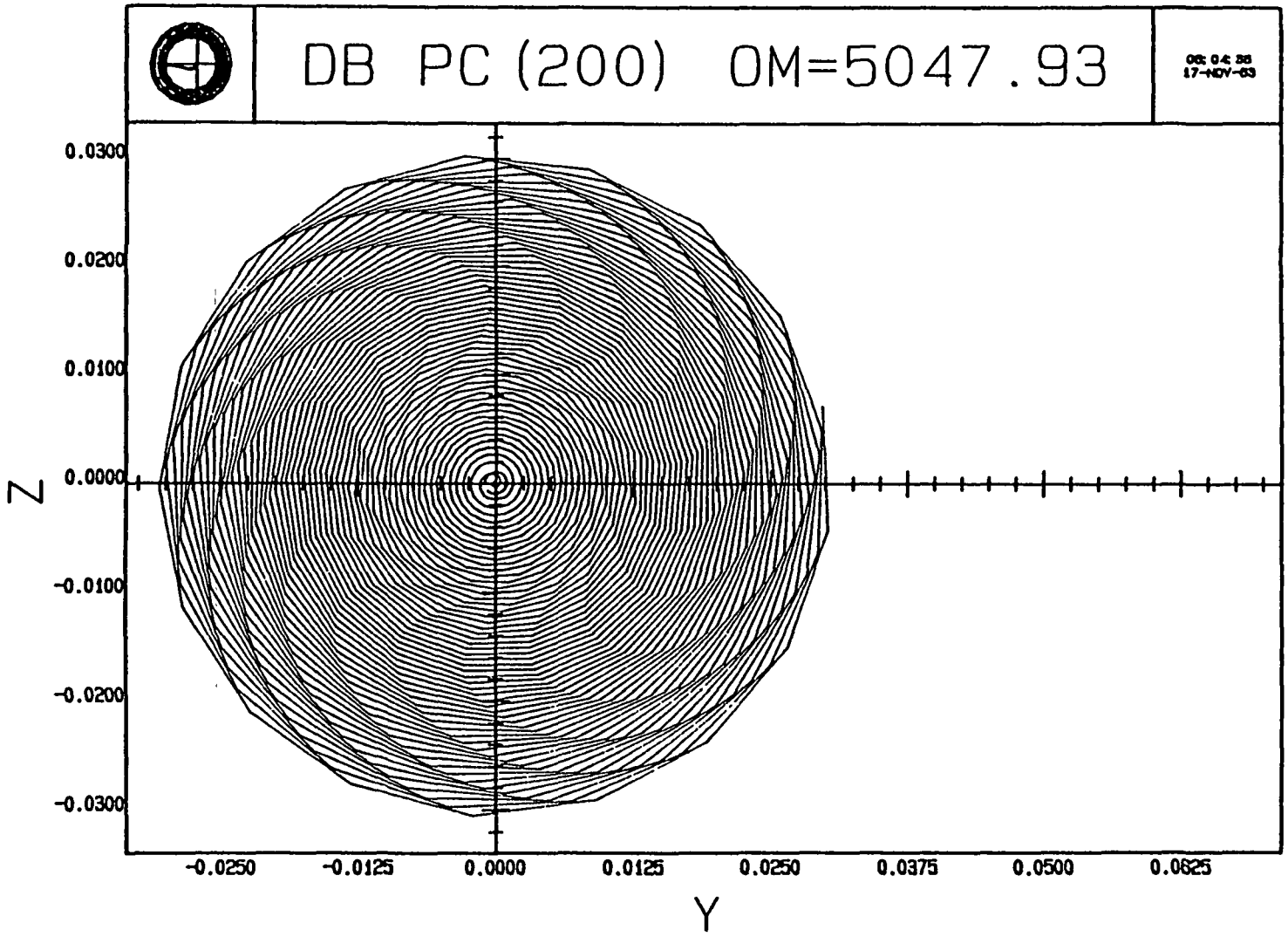

Figure 3.7a Whirl Orbit at the Stability Boundary.

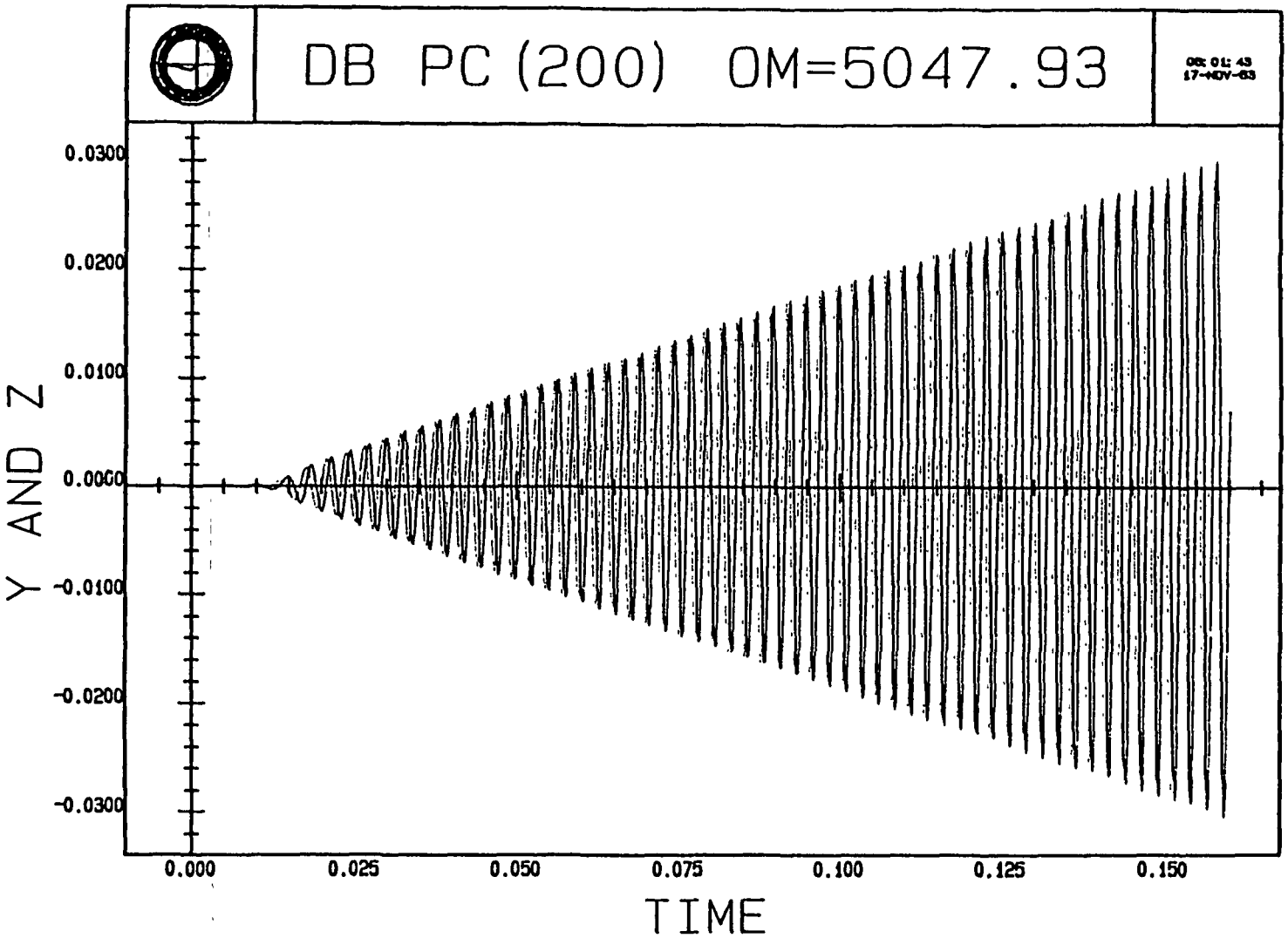

Figure 3.7b The x and y Components as a Function of Time.

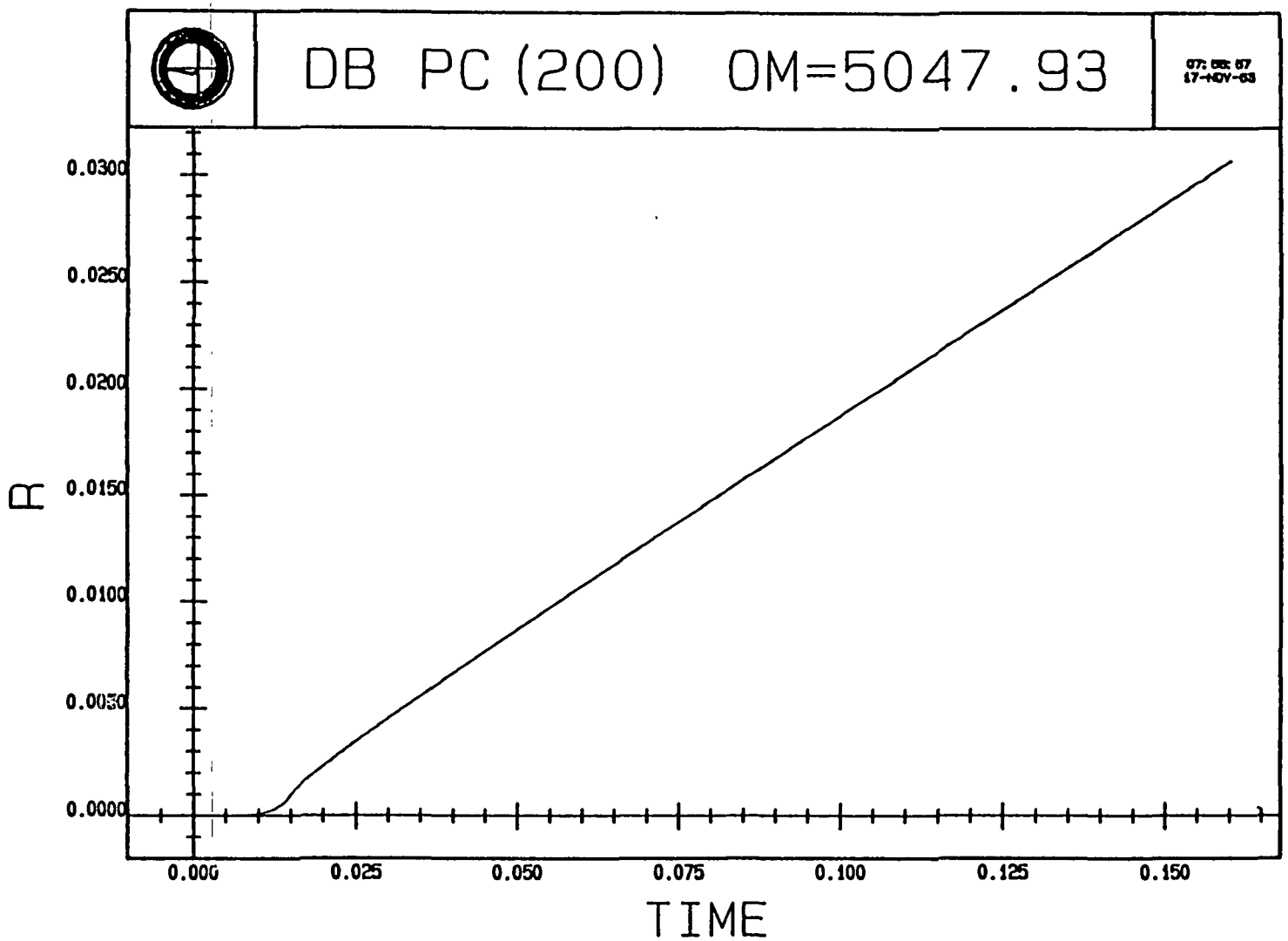

Figure 3.7c The Orbit Radius as a Function of Time.

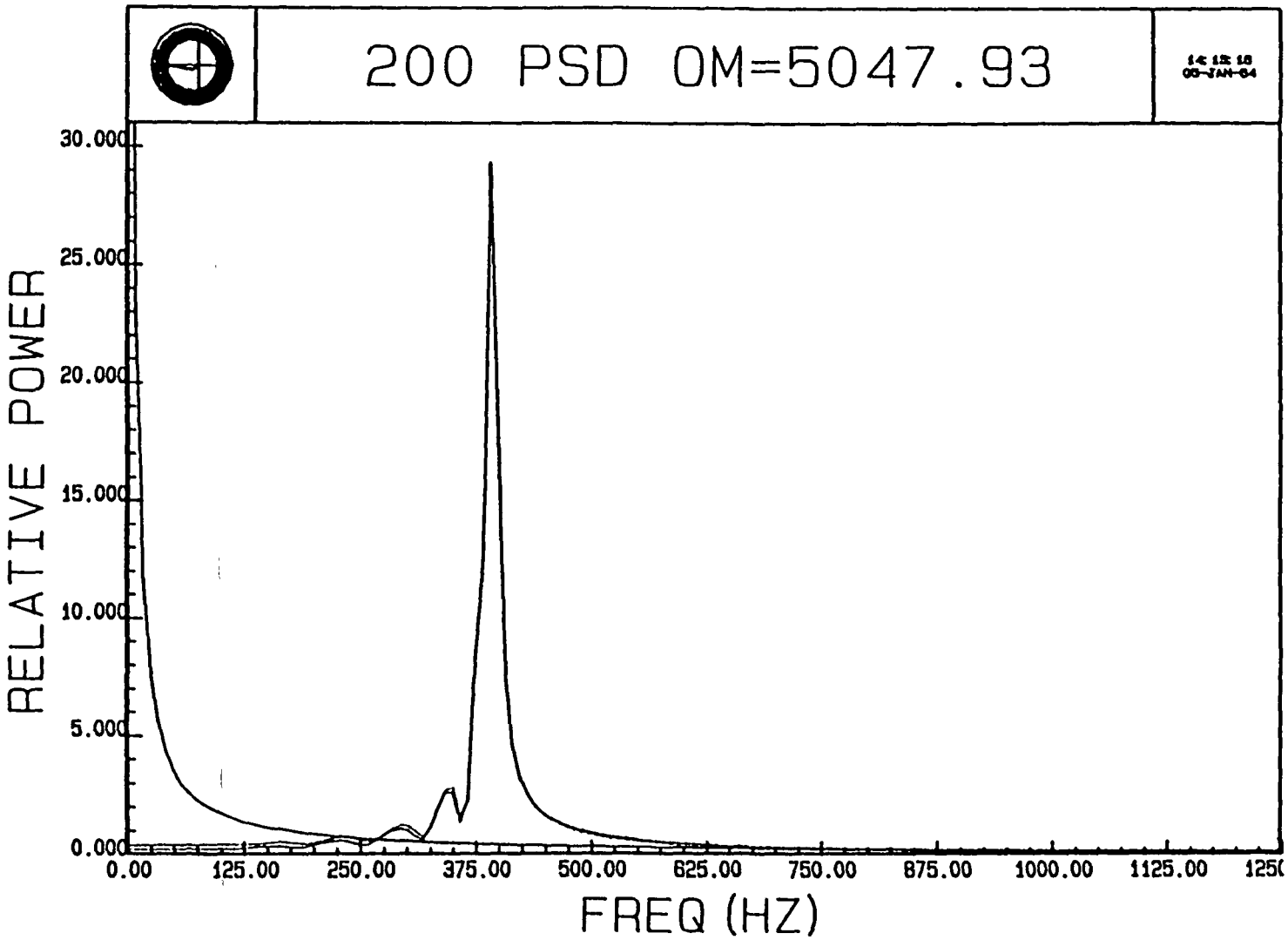

Figure 3.7d PSD of the Orbit; Shaft Speed: 803.4 Hz.

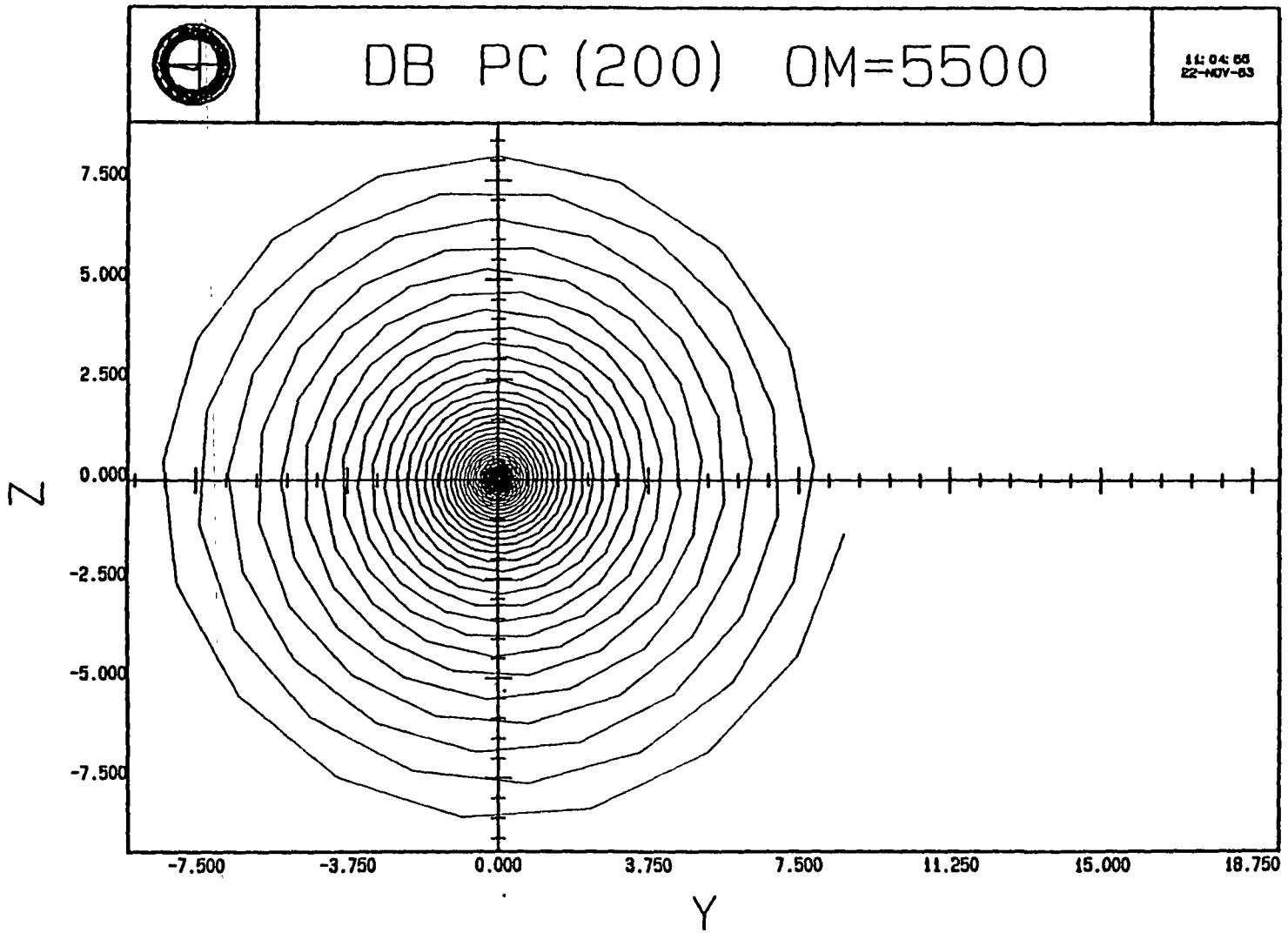

Figure 3.8a Unstable Whirl Orbit.

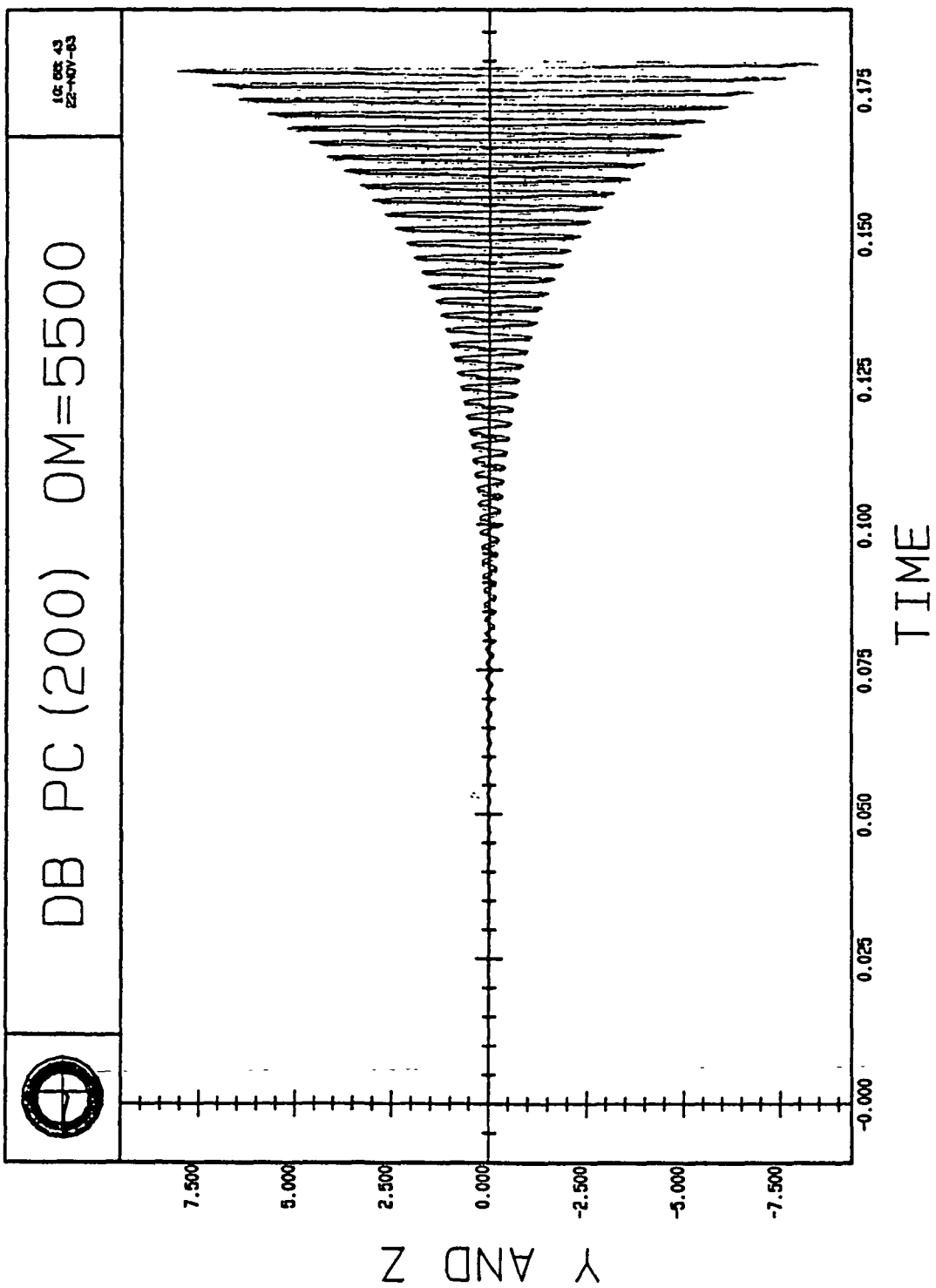

Figure 3.8b The x and y Components as a Function of Time.

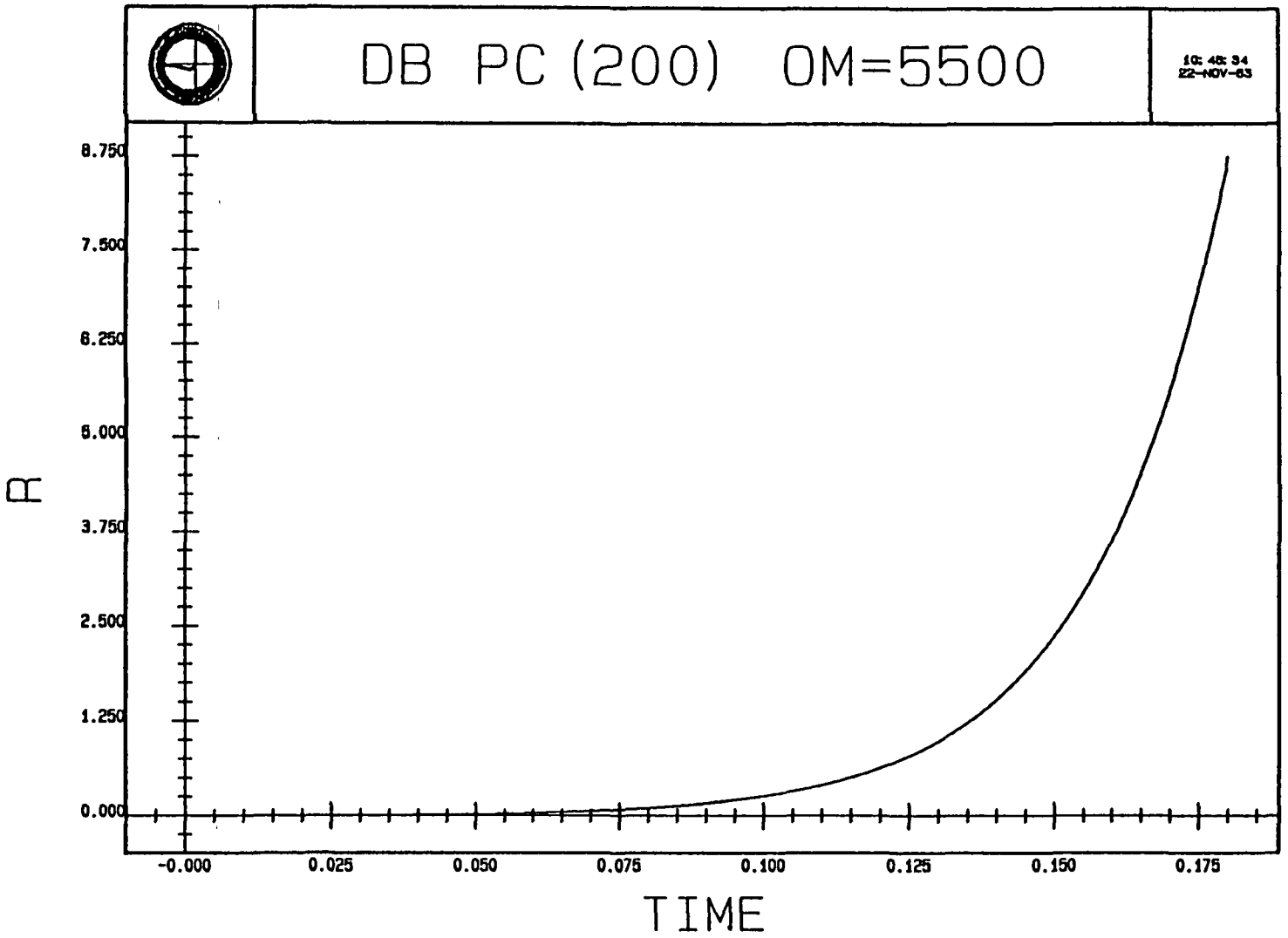

Figure 3.8c The Orbit Radius as a Function of Time.

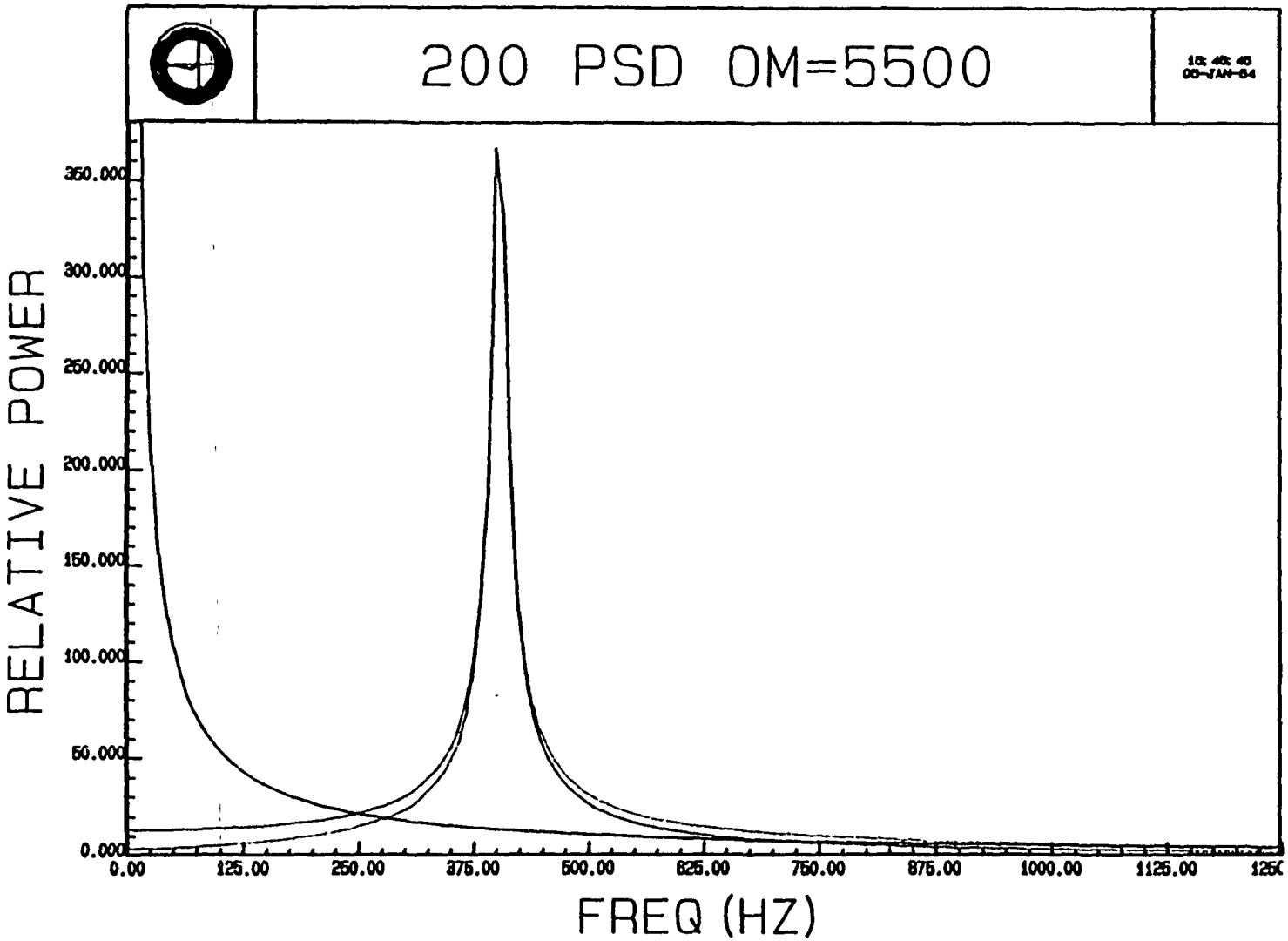

Figure 3.8d PSD of the Unstable Orbit; Shaft Speed: 803.4 Hz.

The state equations are

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{-(K_B + K_S) - C_Q\dot{\phi}_o + m\dot{\phi}_o^2}{m} x_1 - \frac{C_S}{m} x_2 + \frac{2m\dot{\phi}_o r_o - C_Q r_o}{m} x_3 \tag{3.34}$$

$$\dot{x}_3 = \frac{Q_S - C_S\dot{\phi}_o}{m\, r_o} x_1 + \frac{C_Q}{mr_o} x_2 - \frac{C_S}{m} x_3$$

Following a procedure identical to that described in the above paragraphs, the system matrix is formed from which we obtain the system characteristic equation, denoted $P_p(s)$.

$$P_p(s) = s^3 + \frac{2C_S}{m} s^2 + \frac{C_S^2 + C_Q^2 - 2m\dot{\phi}_o C_Q + m(K_B + K_S + C_Q\dot{\phi}_o - m\dot{\phi}_o^2)}{m^2} s + \frac{C_S\,[(K_B + K_S) + C_Q\dot{\phi}_o - m\dot{\phi}_o^2]}{m^2} \tag{3.35}$$

The absence of the variable ϕ from the expressions in polar coordinates reduces the system order to three. Subsequent examination of the characteristic equation in 3.35 indicates the instability frequency is 5047.93, further verifying the analysis done thus far.

3.3 NONLINEAR SYSTEM WITH IMBALANCE

The next step in the analysis is to consider the system with a deadband and an imbalance force driving the whirl. As before, an equilibrium radius may be determined. Figure 3.9 is the force diagram that is used to determine the value of r_o. The conditions for equilibrium are that the forces in the radial direction must sum to zero, as well as those in the tangential direction, leaving only the shaft spin rate to drive a synchronous C-type whirl with constant radius. The equations for the radial and tangential forces are given in Equations 3.36 and 3.37, respectively.

$$(Kr - K_{Bg}) + C_Q\,\omega r - m\,\omega^2\varepsilon_r - m\,\omega^2 r = 0 \tag{3.36}$$

$$Q_S r - C_S\omega r + m\,\omega^2\varepsilon_\phi = 0 \tag{3.37}$$

Note that $K = K_B + K_S$. Because

$$\varepsilon_r = \varepsilon \cos\omega t \tag{3.38}$$

and

$$\varepsilon_\phi = \varepsilon \sin\omega t, \tag{3.39}$$

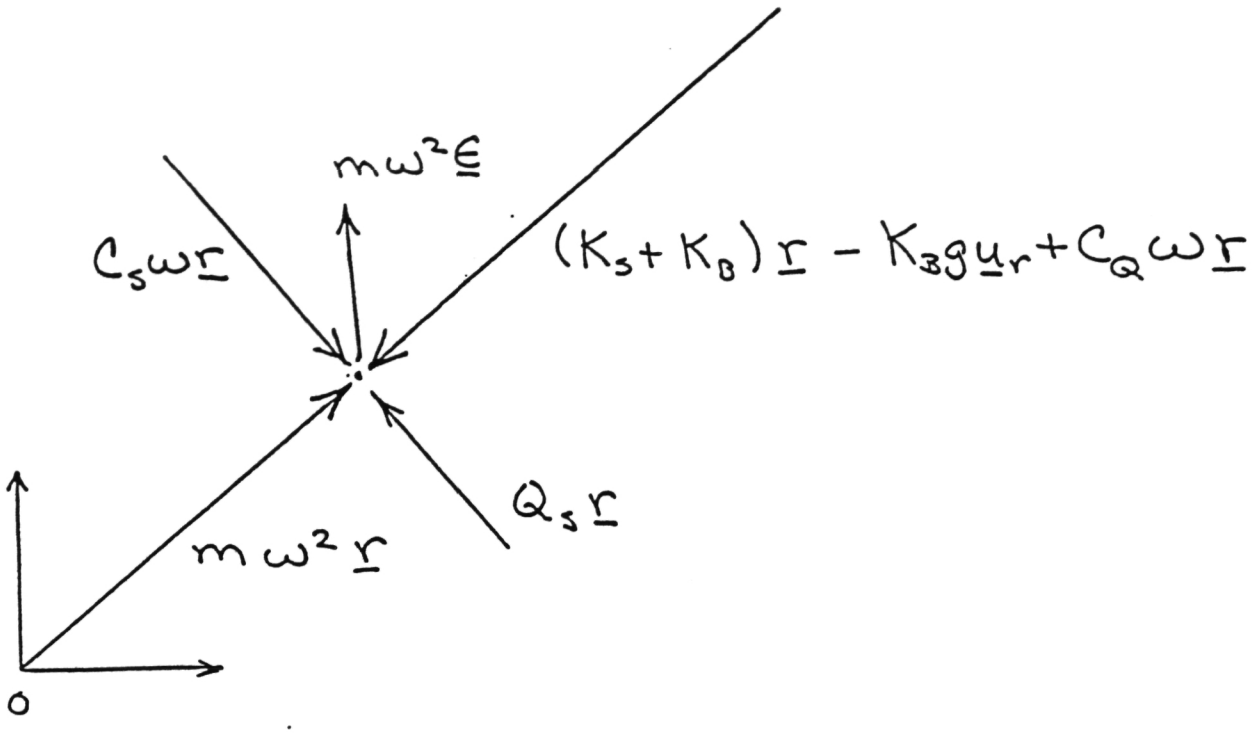

Figure 3.9 Force Diagram Used to Determine r_0.

we have

$$\varepsilon^2 = \varepsilon_r^2 + \varepsilon_\phi^2 \quad . \tag{3.40}$$

From Equation 3.36,

$$\varepsilon_r = \frac{[(K + C_Q \omega) - m\omega^2]r - K_B g}{m\omega^2} \quad . \tag{3.41}$$

Likewise, Equation 3.37 may be solved for ε_ϕ

$$\varepsilon_\phi = -\frac{(Q_S - C_S\omega)r}{m\omega^2} \quad . \tag{3.42}$$

When Equations 3.41 and 3.42 are squared and inserted into Equation 3.40, a quadratic in r is produced.

$$\begin{aligned}[(K + C_Q\omega - m\omega^2)^2 + (Q_S - C_S\omega)^2]r^2 - 2K_B g\,[K + C_Q\omega - m\omega^2]r \\ + [K_B^2 g^2 - \varepsilon^2 m^2 \omega^4] = 0\end{aligned} \tag{3.43}$$

We make use of the quadratic formula at this point to acquire the roots to the above equation.

$$r_{1,2} = \frac{K_B g\,(K + C_Q\omega - m\omega^2) \pm \sqrt{m^2\omega^4\varepsilon^2\,[(K + C_Q\omega - m\omega^2)^2 + (Q_S - C_S\omega)^2 - K_B^2 g^2\,(Q_S - C_S\omega)^2]}}{[(K + C_Q\omega - m\omega^2)^2 + (Q_S - C_S\omega)^2]} \tag{3.44}$$

In order for an equilibrium radius to exist, there must be a positive real solution for r. Close scrutiny of Equation 3.44, particularly the radical term, along with some knowledge of the relative magnitudes of the parameters involved, reveals that the solutions for r will always be real; however, they need not always be positive. In the absence of deadband, except at zero frequency, there is always a positive solution for r for any positive value of ε. Shown in Figure 3.10 is a plot of an orbit resulting from a simulation executed with a zero dead-band. Note from the PSD that this is a synchronous whirl.

However, for non-zero values of g, there are combinations of values of ε and g for which there is no positive solution to r_o. The relationship which must hold between ε and g for an equilibrium radius to exist is

$$\frac{g}{\varepsilon} < \frac{m\omega^2}{K_B} \quad . \tag{3.45}$$

This relationship is dictated by Equation 3.44. As indicated earlier, the range of values of interest in this study are from 0.5 mils to 2.5 mils for the deadband and from 0.1 to 0.2 mils for the rotor eccentricity. Therefore, the minimum frequency at which an r_o will exist is 3498.8 radians per second, a frequency greater than the normal operating speed of 3141.59 radians/second used in most of the runs executed in our analysis. The curves shown in Figure 3.11

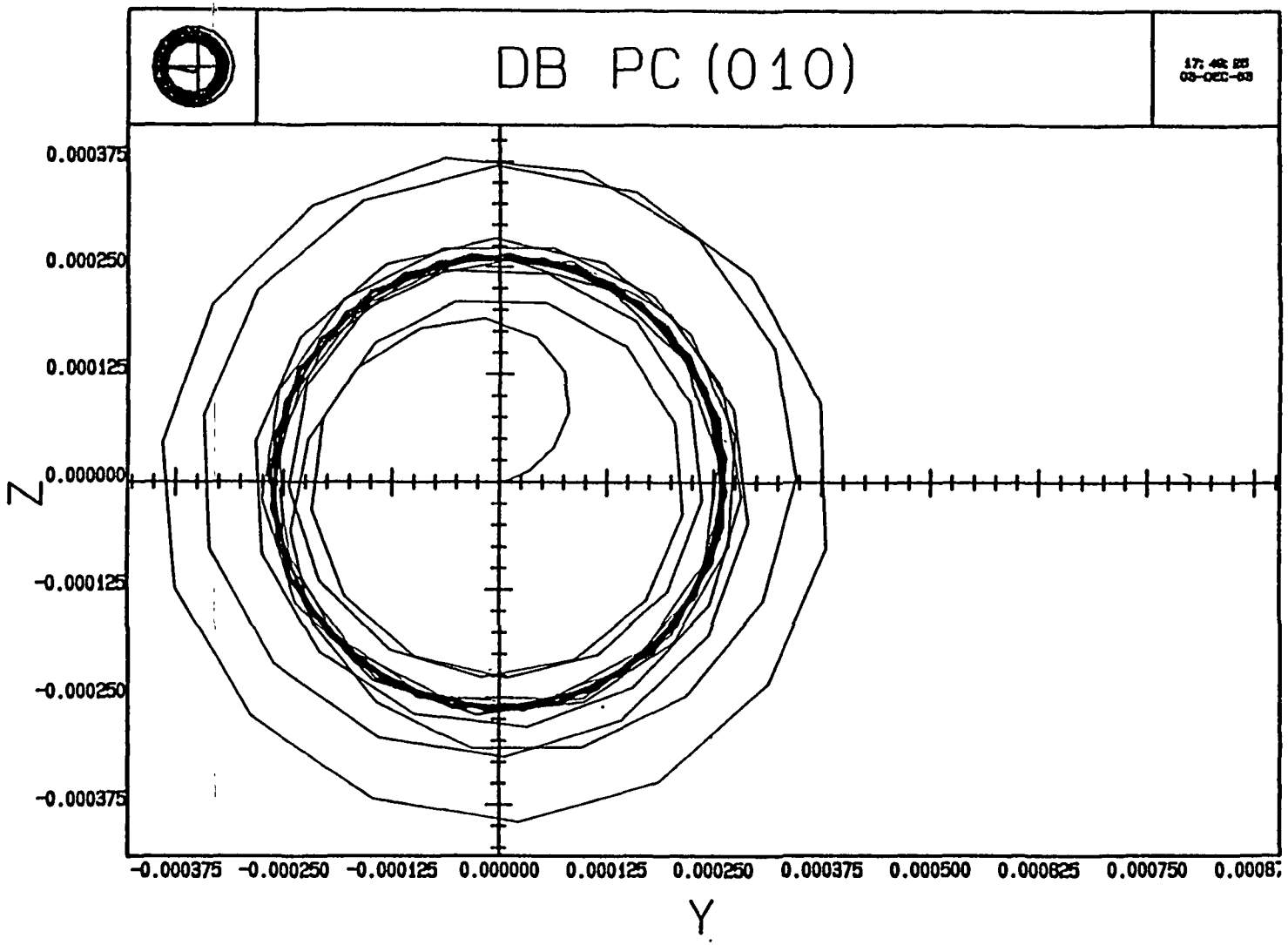

Figure 3.10a Synchronous Orbit Including Transients.

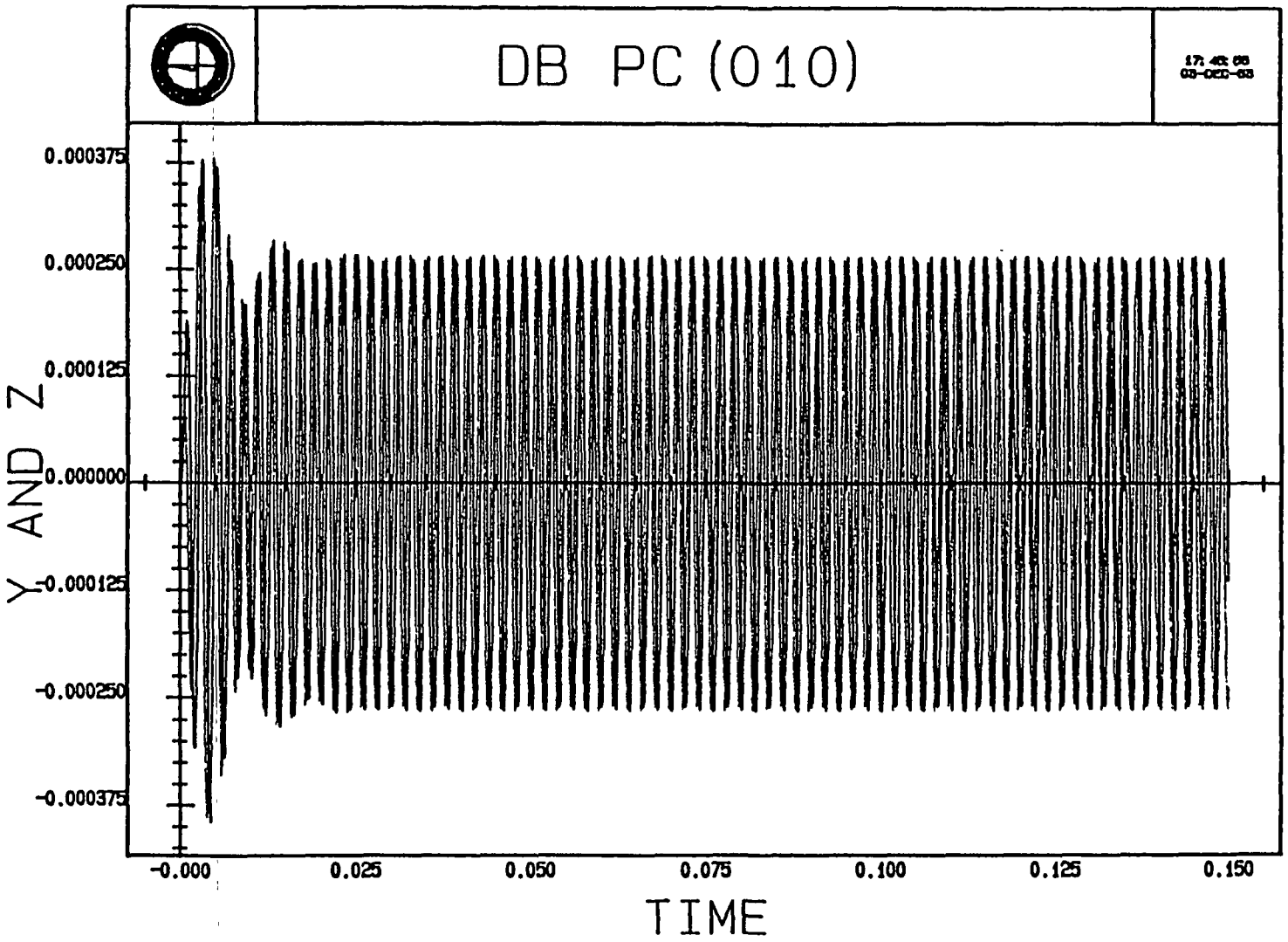

Figure 3.10b The x and y Components as a Function of Time.

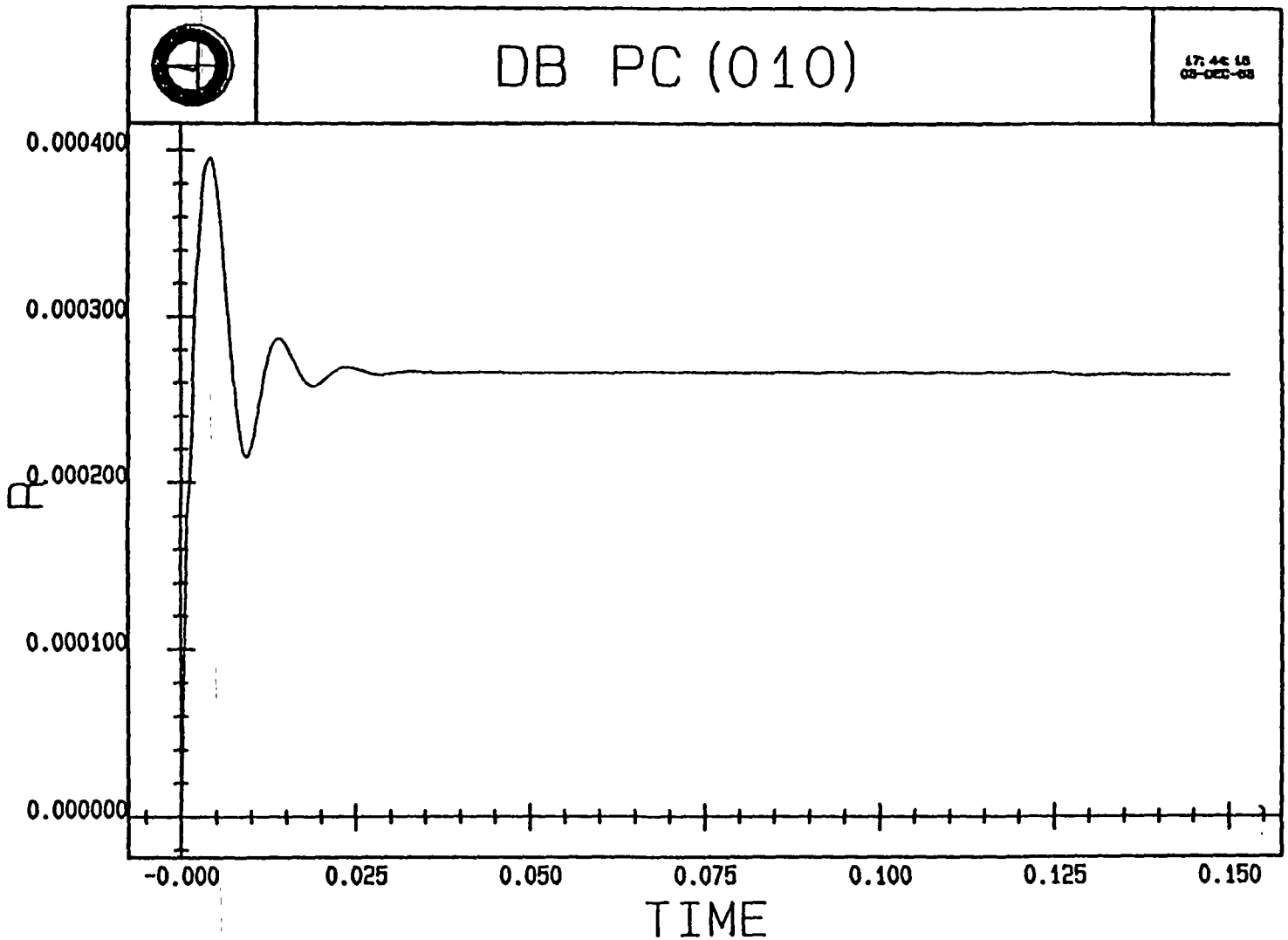

Figure 3.10c The Orbit Radius as a Function of Time.

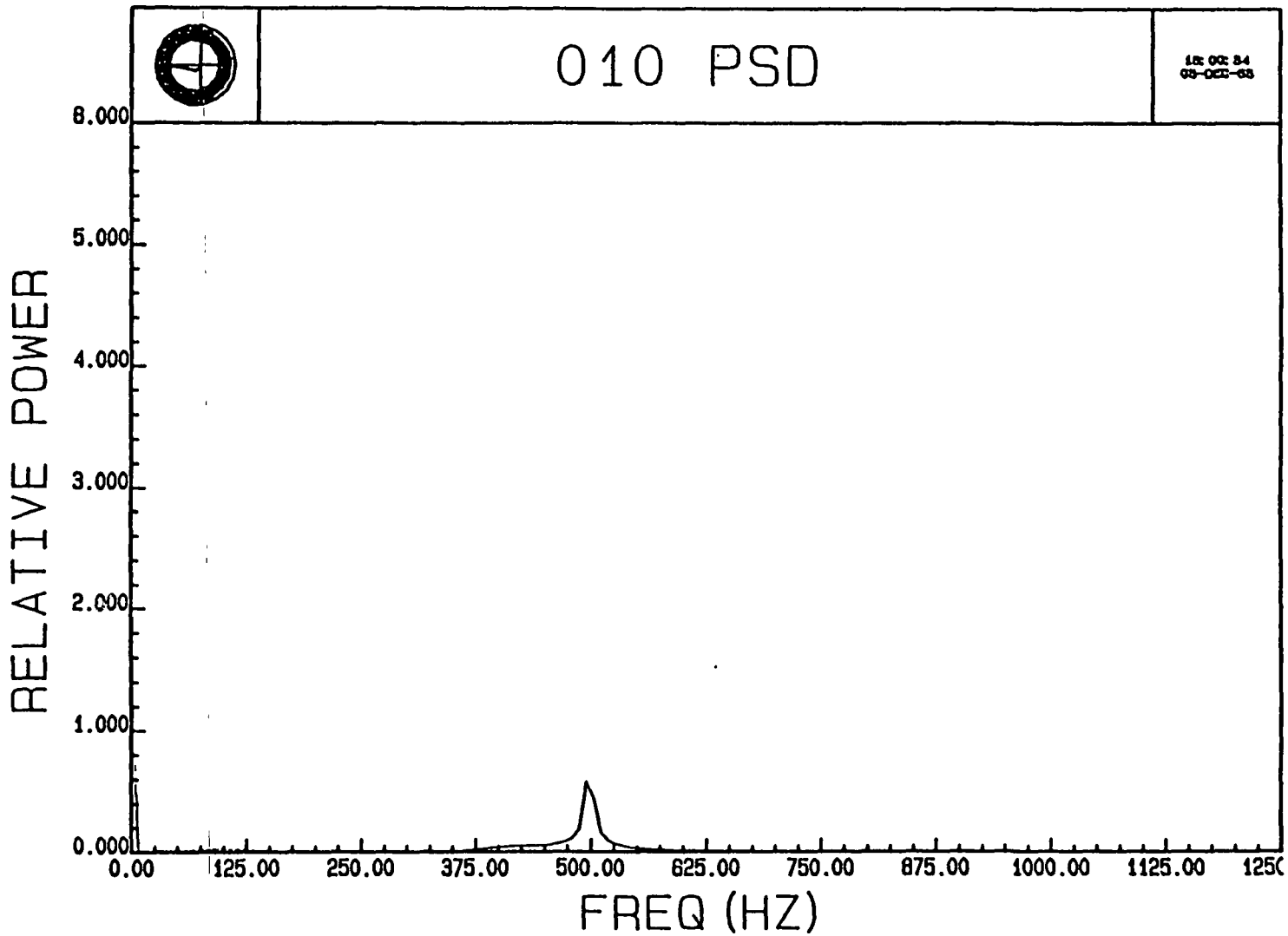

Figure 3.10d PSD of the Synchronous Orbit; Shaft Speed: 500 Hz.

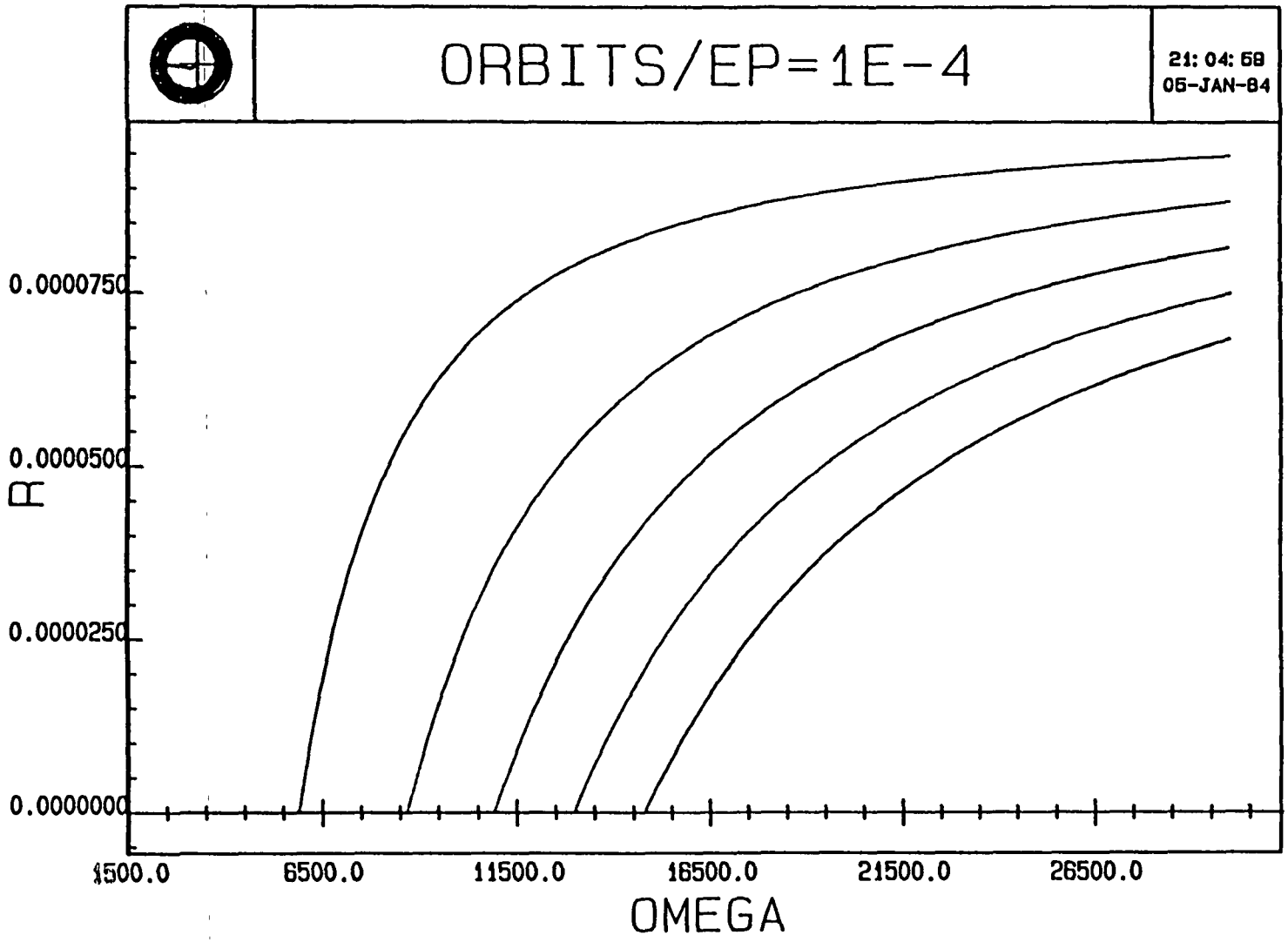

Figure 3.11a Orbital Radii as a Function of Frequency, EP = 0.1 Mil.

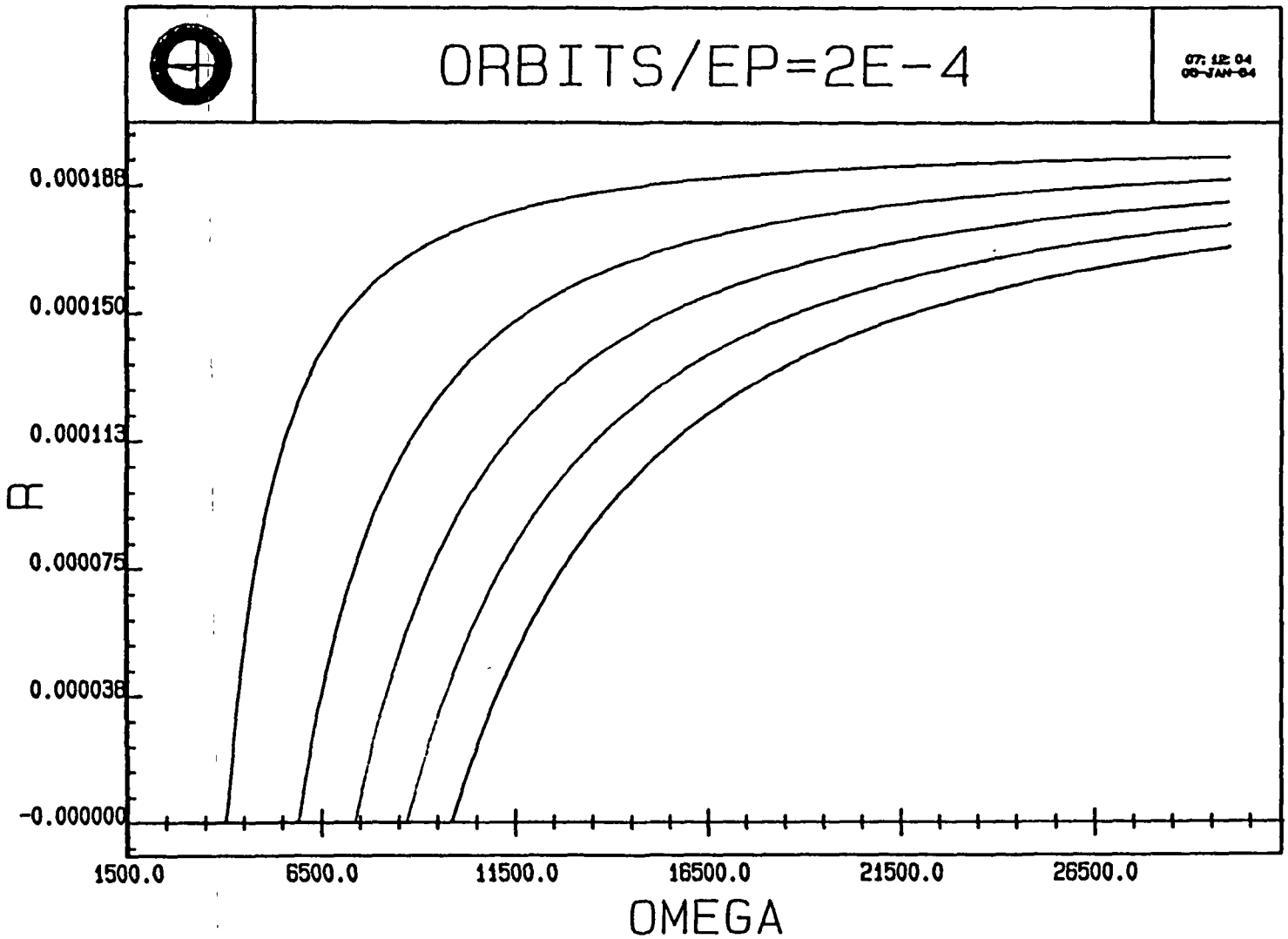

Figure 3.11b Orbital Radii as a Function of Frequency, EP = 0.2 Mil.

indicate the equilibrium orbit radii as a function of frequency for the deadbands and imbalances indicated above. The whirl radii are much less than that of the deadband; therefore, no bearing load results. The requencies at which these exist, however, are much above the stability boundary.

For our studies, what happens with regard to the orbits seen when the simulation is executed at the normal operating speed is that a variety of B and C-type motions are present. In many cases, subsynchronous C-type limit cycles are seen. Figures 3.12 and 3.13 show B-type orbits with Figure 3.14 depicting a subsnychronous C-type motion. The orbit in Figure 3.14 is not an equilibrium whirl, it is a tiny bit offset from the center.

Analysis of the stability properties of this system is carried out in a manner quite similar to that in the previous section. We wish to linearize about the equilibrium radius and determine the stability boundary. Again, we chose to use the cartesian coordinates formulation. Refer back to Figure 3.4 to refresh your memory on the definitions of the unit vectors $\underline{e}_y$ and $\underline{e}_z$. The differential equations are

$$\ddot{y} = \frac{-K}{m} y - \frac{C_S}{m} \dot{y} - \frac{Q_S}{m} z - \frac{C_Q}{m} \dot{z} + \omega^2 \varepsilon \cos \omega t \qquad (3.46)$$

$$\ddot{z} = \frac{-K}{m} z - \frac{C_S}{m} \dot{z} + \frac{Q_S}{m} y + \frac{C_Q}{m} \dot{y} + \omega^2 \varepsilon \sin \omega t \qquad (3.47)$$

Inspection of the above reveals that the only difference between these differential equations and Equations 3.19 and 3.20 is the addition of the shaft eccentricity term. No system state variable appears in this term, therefore it acts only as a forcing function and has no effect on the characteristic equation for the system. Hence, the stability properties of the rotor model with deadband and imbalance are the same as those for the model with only deadband included. That is, with regard to the stability boundary.

3.4 SIDE FORCE CONSIDERATIONS.

When a side force is included in the system analysis, we must rethink the way the investigation is conducted. We cannot simply add the side force term into the equations derived thus far because the behavior of the system changes. In addition to that consideration, adding the side force term to the equations describing the model in Section 3.3 makes for a rather difficult time when analysis is attempted. A look back at Equations 2.13 and 2.14 should make this assertion obvious. Transcendental equations result when an effort is made to determine the equilibrium conditions. The approach, therefore, will be to consider the simple model and include the effects of deadband and side forces, temporarily omitting the imbalance considerations.

Under the influence of a side force, the rotor shifts to a position of equilibrium, a single point rather than an orbit, that is, with no imbalance present. The effect of an imbalance term, in general, is that the rotor whirls about the new equilibrium point. The position of the equilibrium point is dependent upon the magnitude of the side force, the deadband, and the stiffness coefficients. The position of the equilibrium point, with respect to the deadband, determines the type of orbit in which the rotor will whirl.

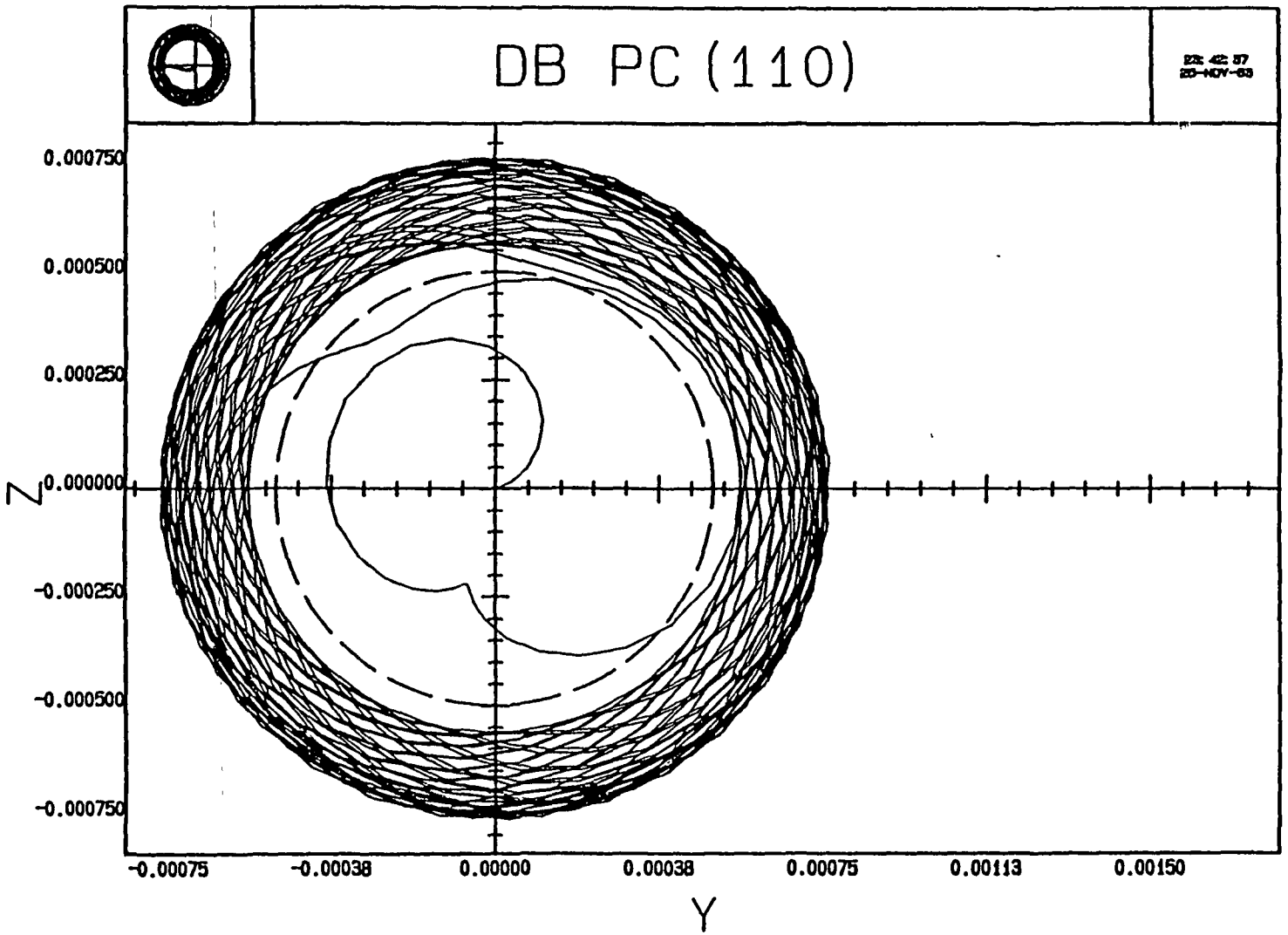

Figure 3.12a B-Type Motion Including Transients.

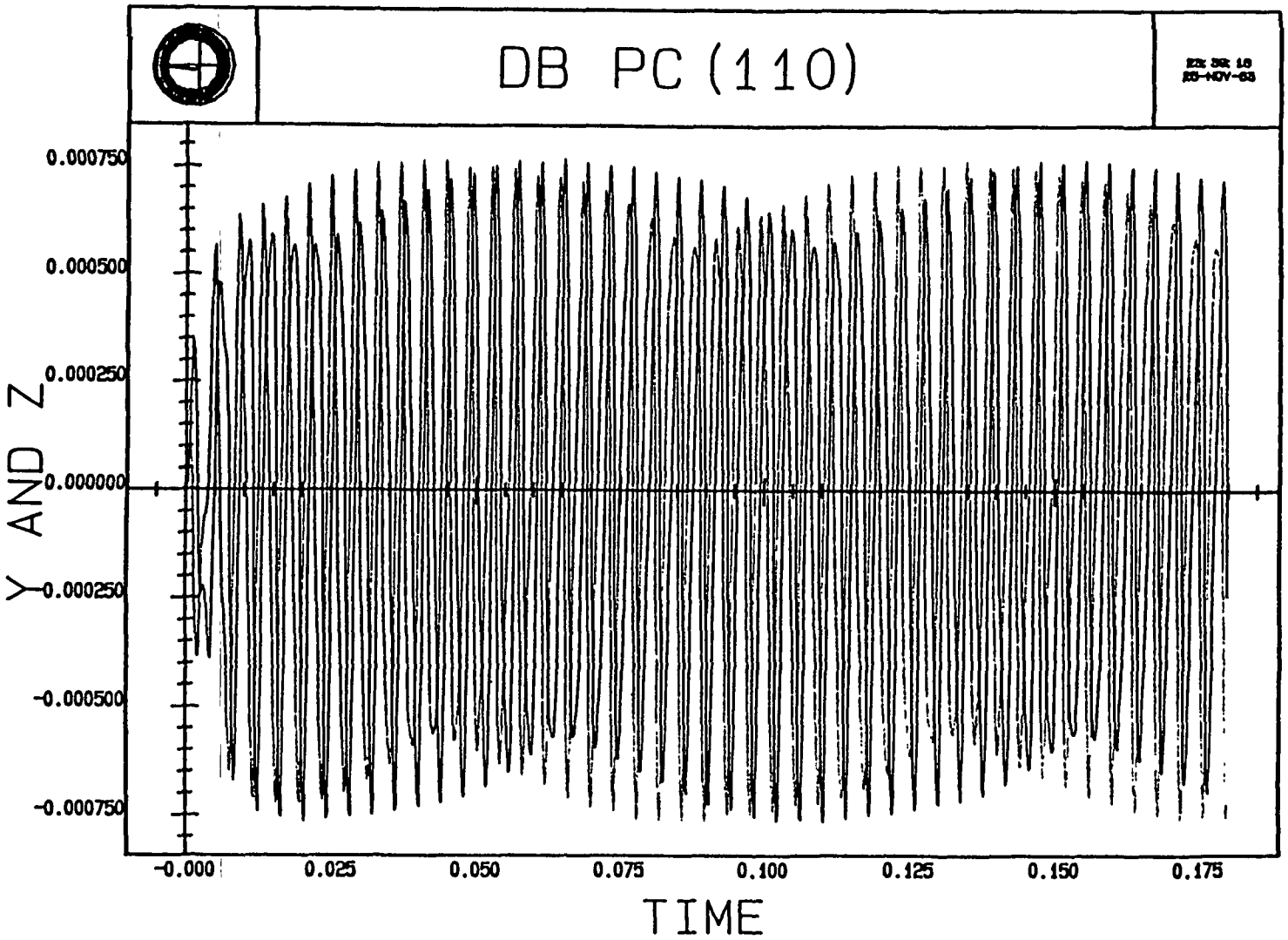

Figure 3.12b The x and y Components as a Function of Time.

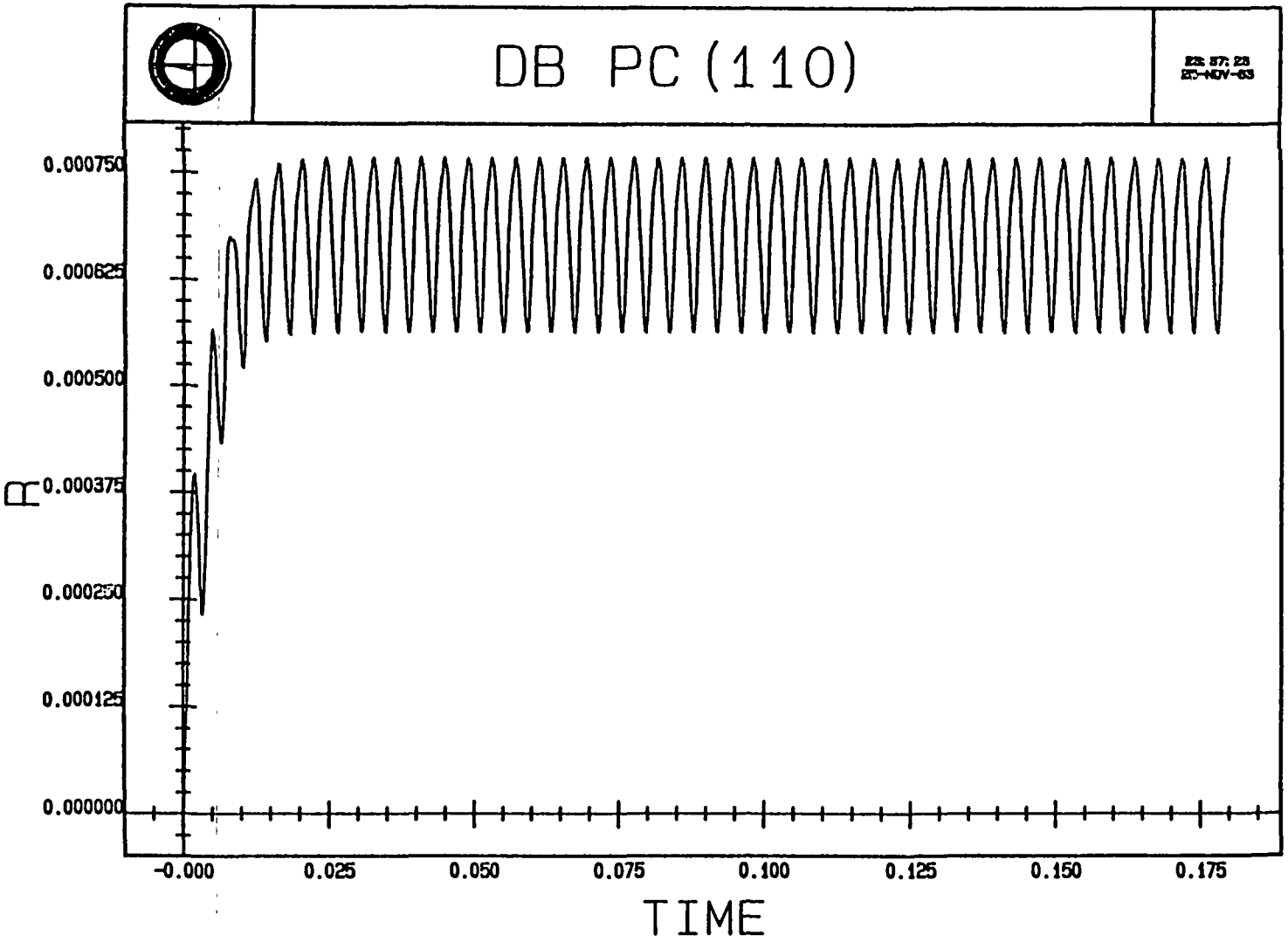

Figure 3.12c The Orbit Radius as a Function of Time.

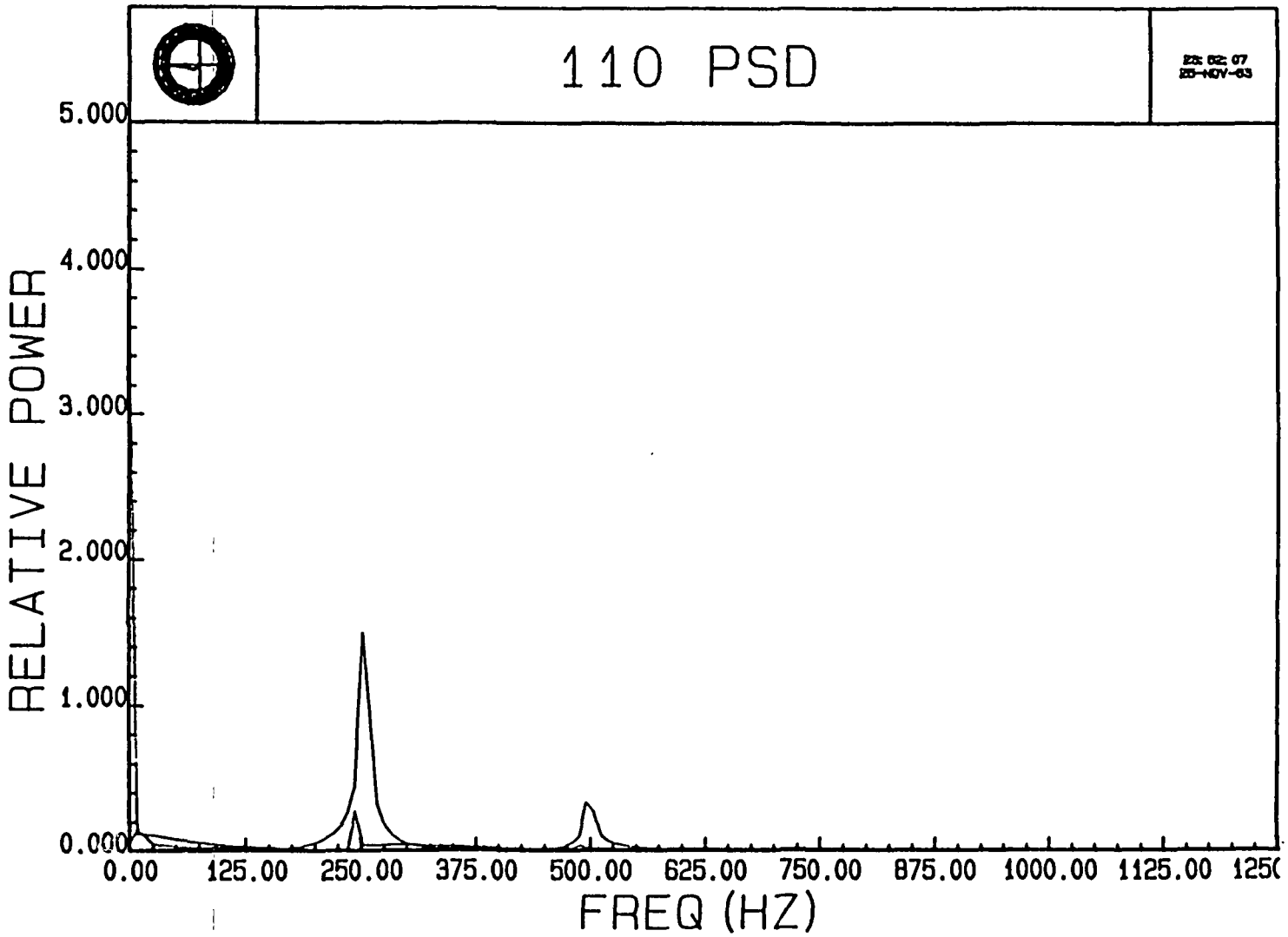

Figure 3.12d PSD of the B-type Orbit; Shaft Speed: 500 Hz.

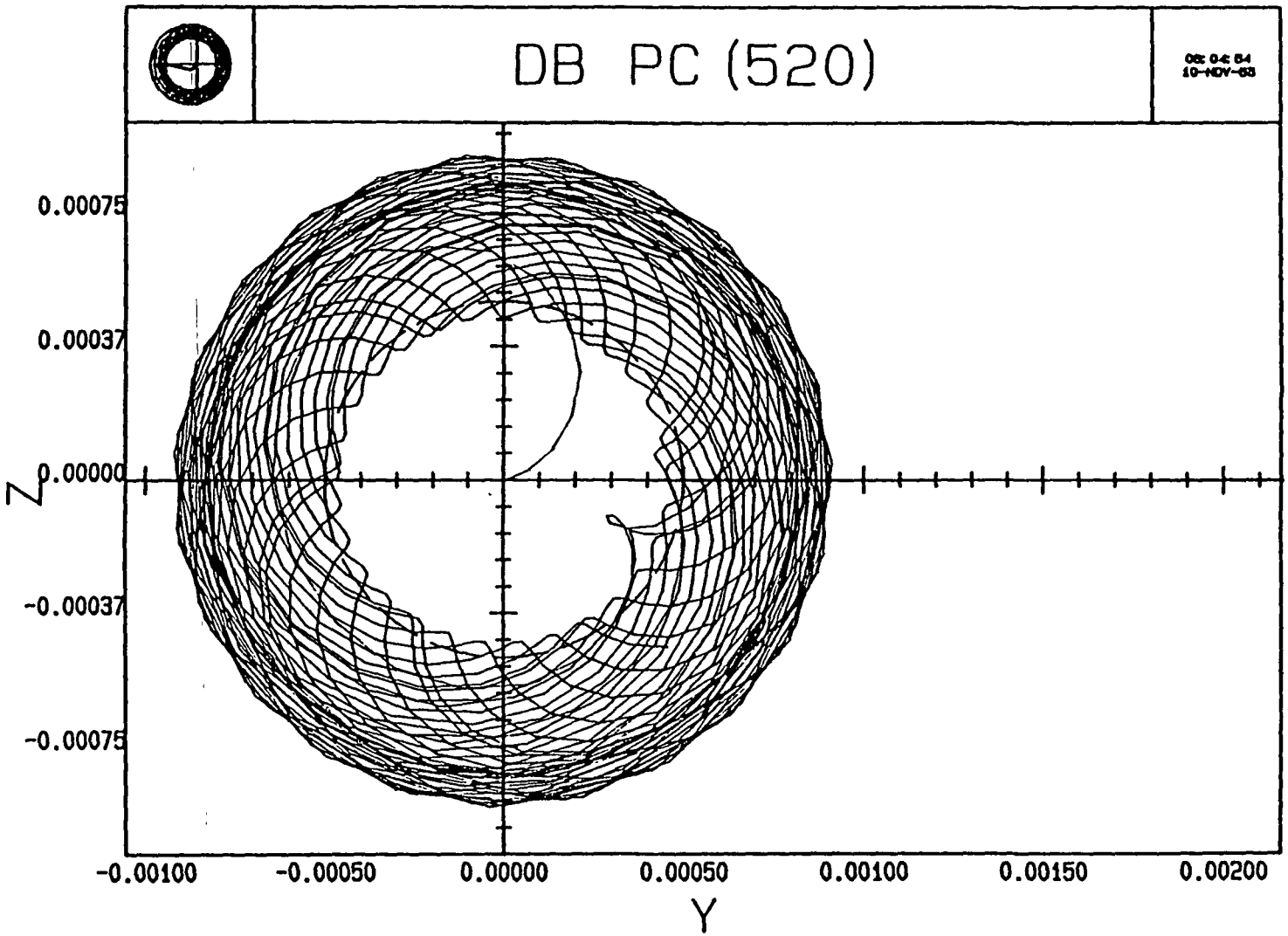

Figure 3.13a B-Type Motion Including Transients.

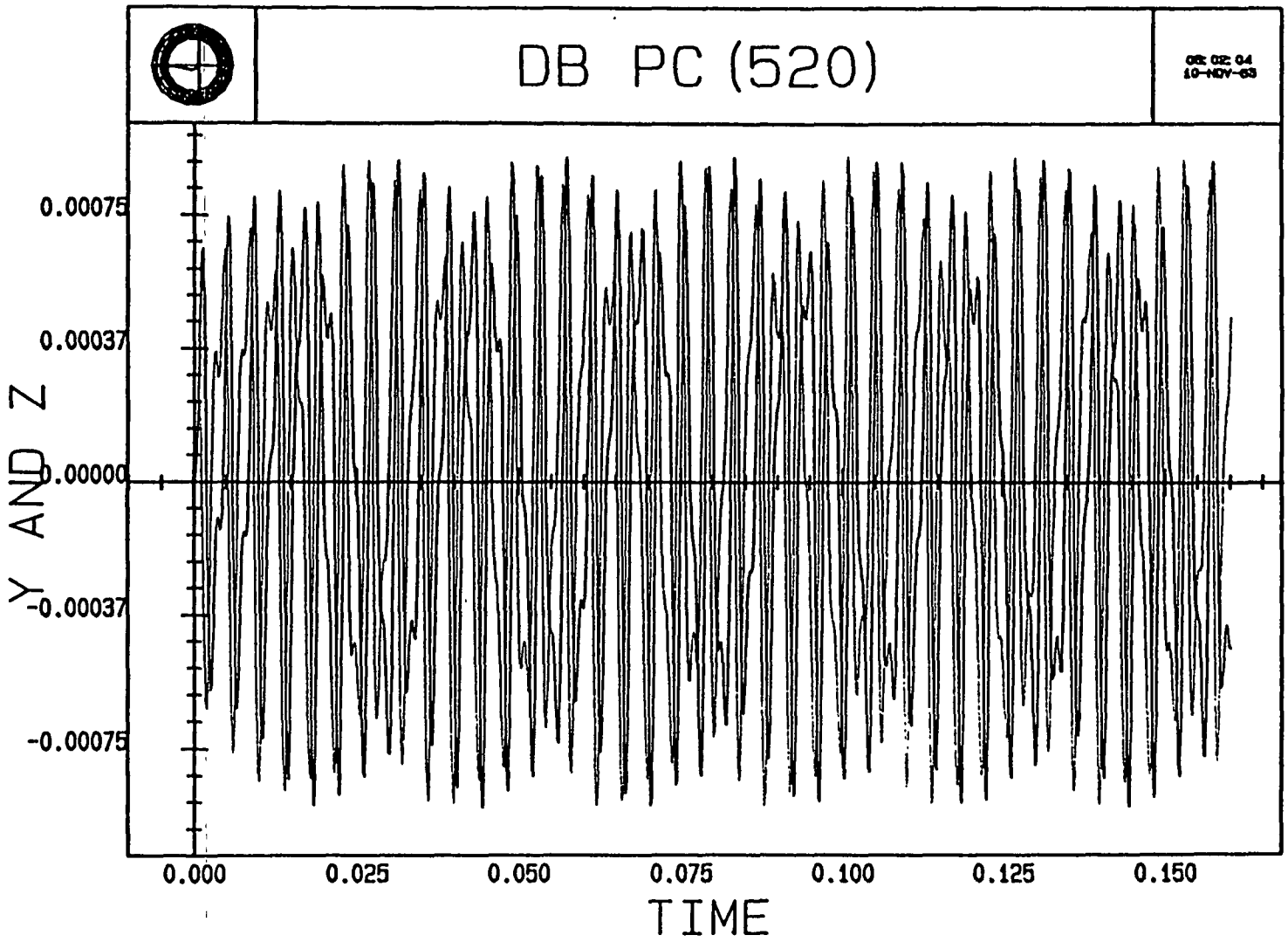

Figure 3.13b The x and y Components as a Function of Time.

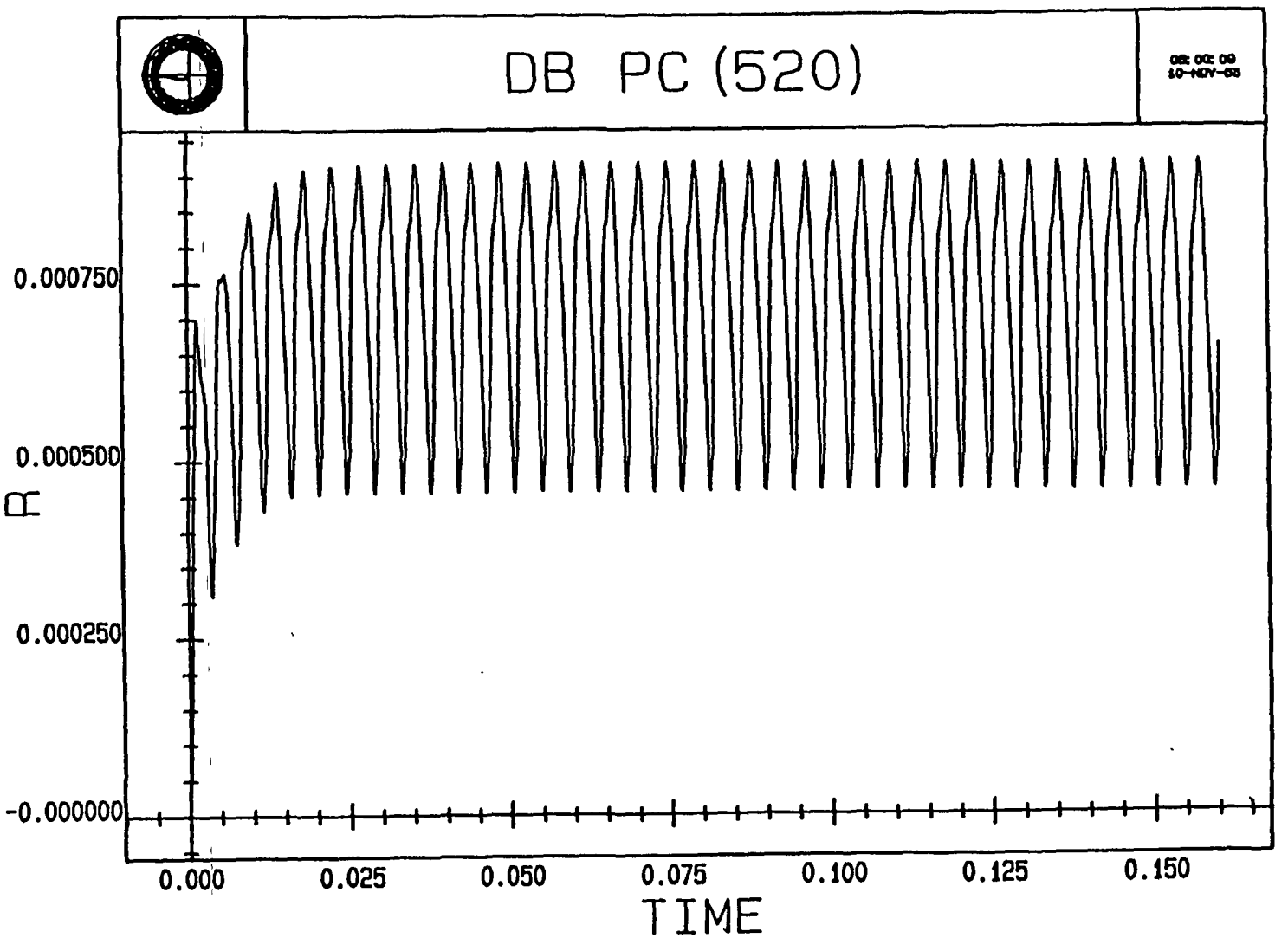

Figure 3.13c The Orbit Radius as a Function of Time.

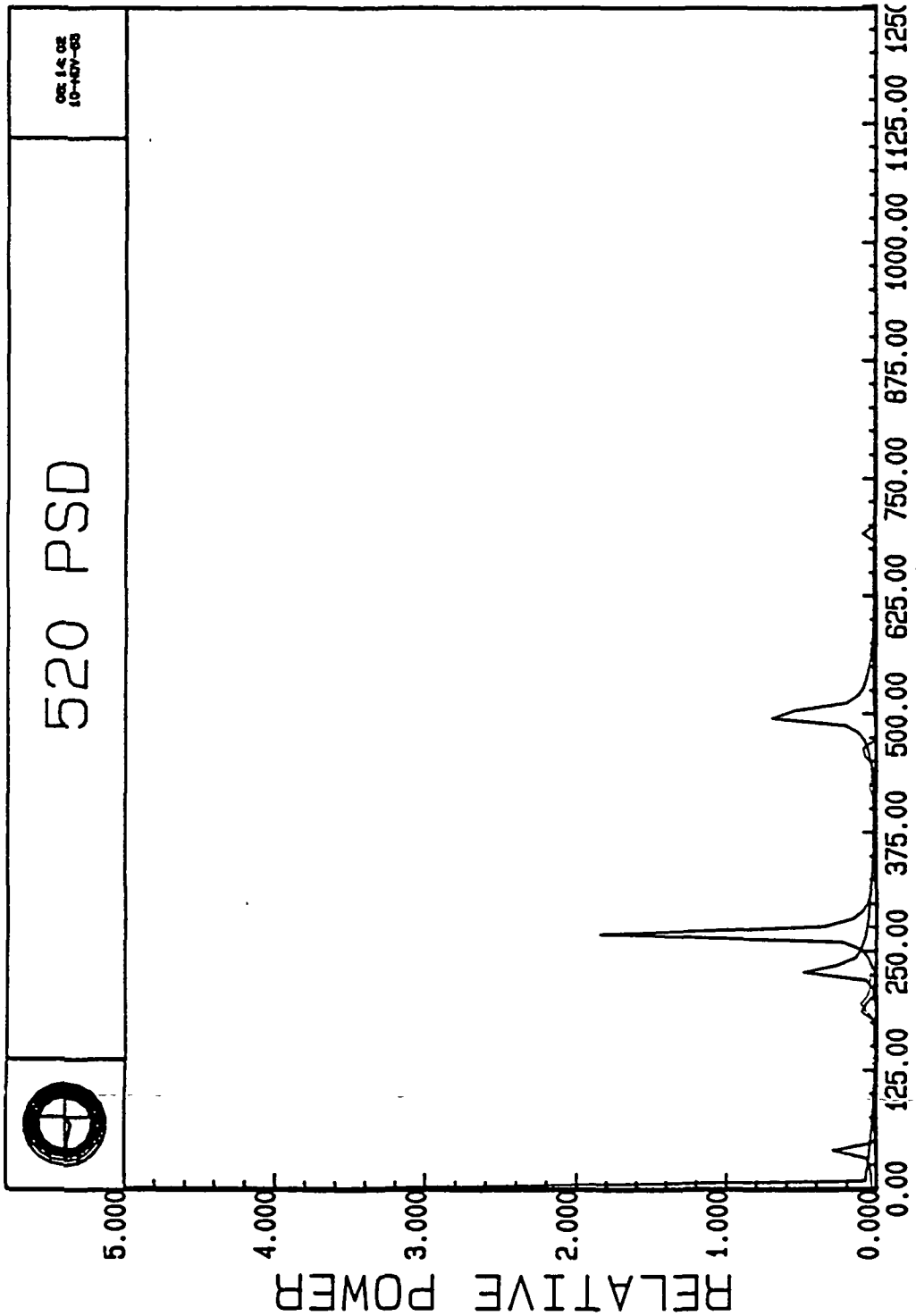

Figure 3.13d PSD of the B-Type Orbit; Shaft Speed: 500 Hz.

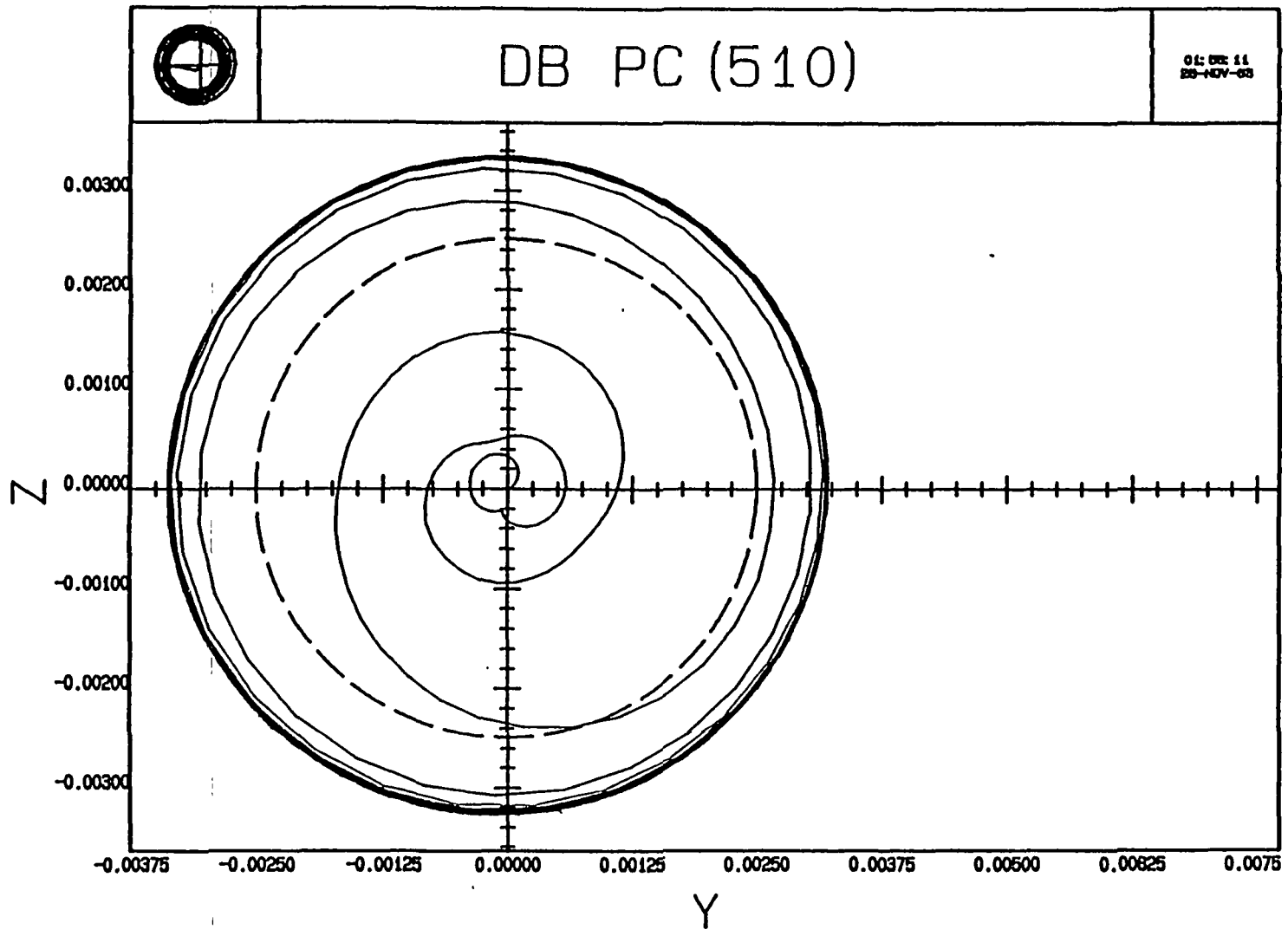

Figure 3.14a Subsynchronous C-Type Whirl Including Transients.

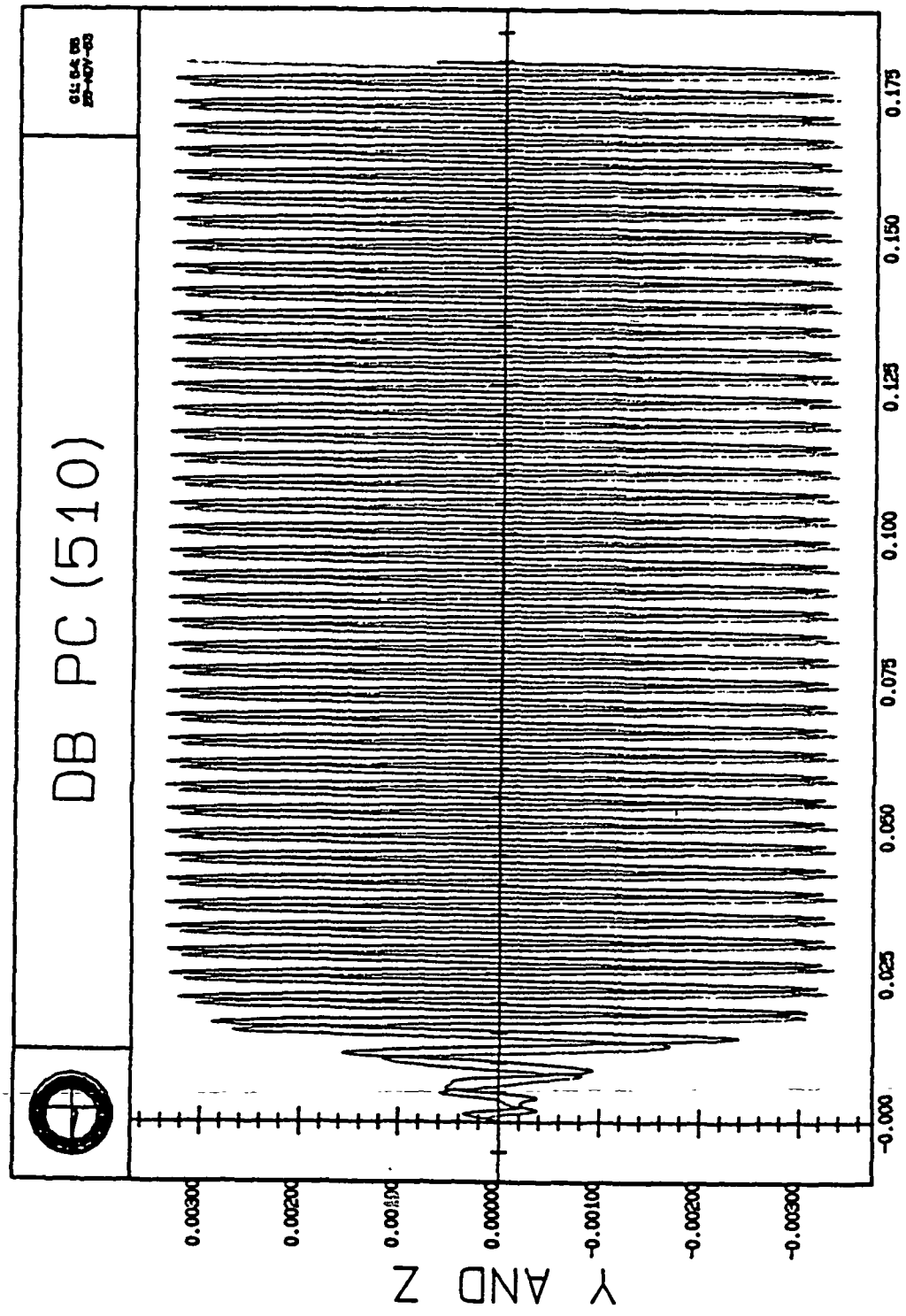

Figure 3.14b The x and y Components as a Function of Time.

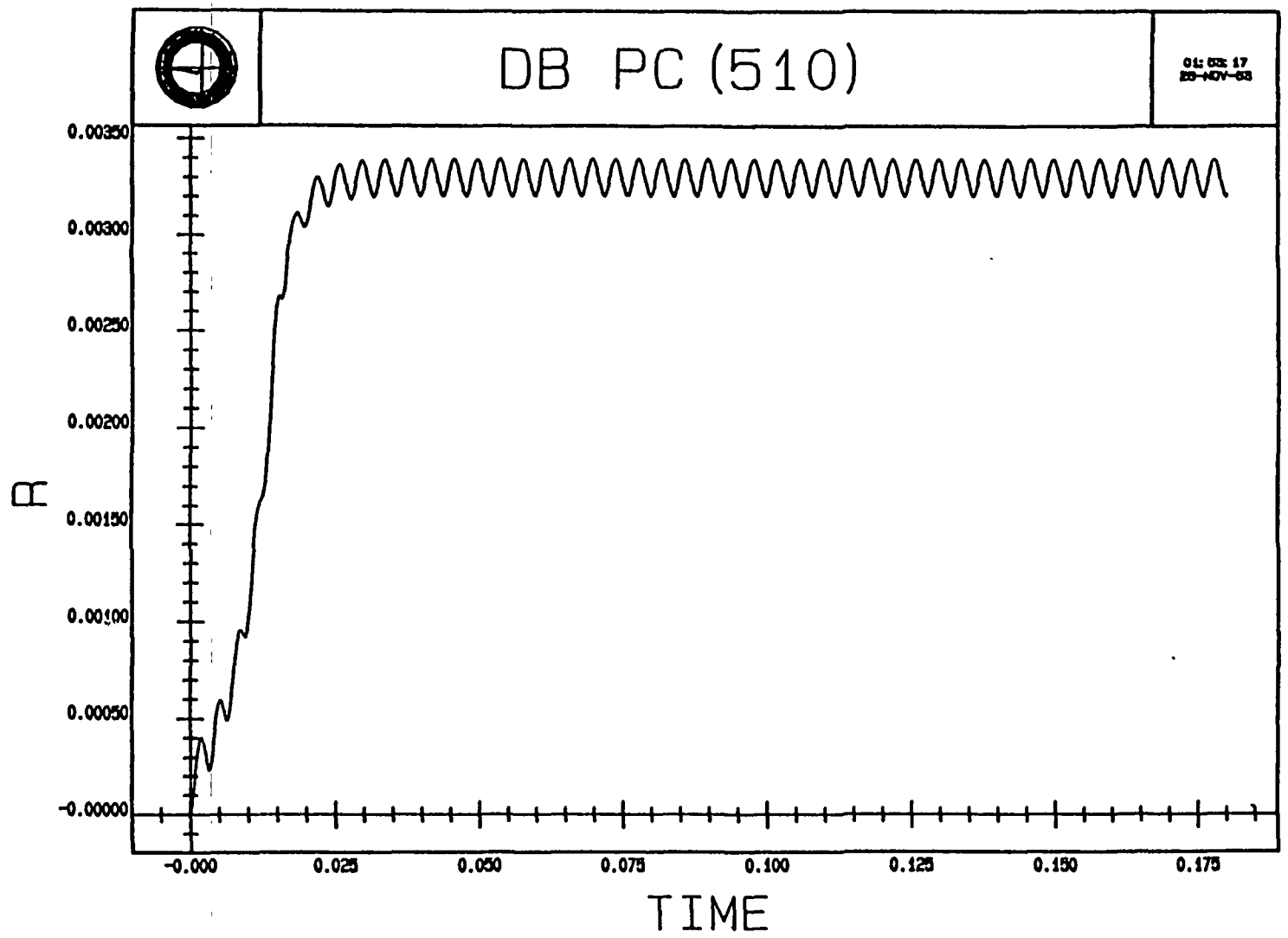

Figure 3.14c The Orbit Radius as a Function of Time.

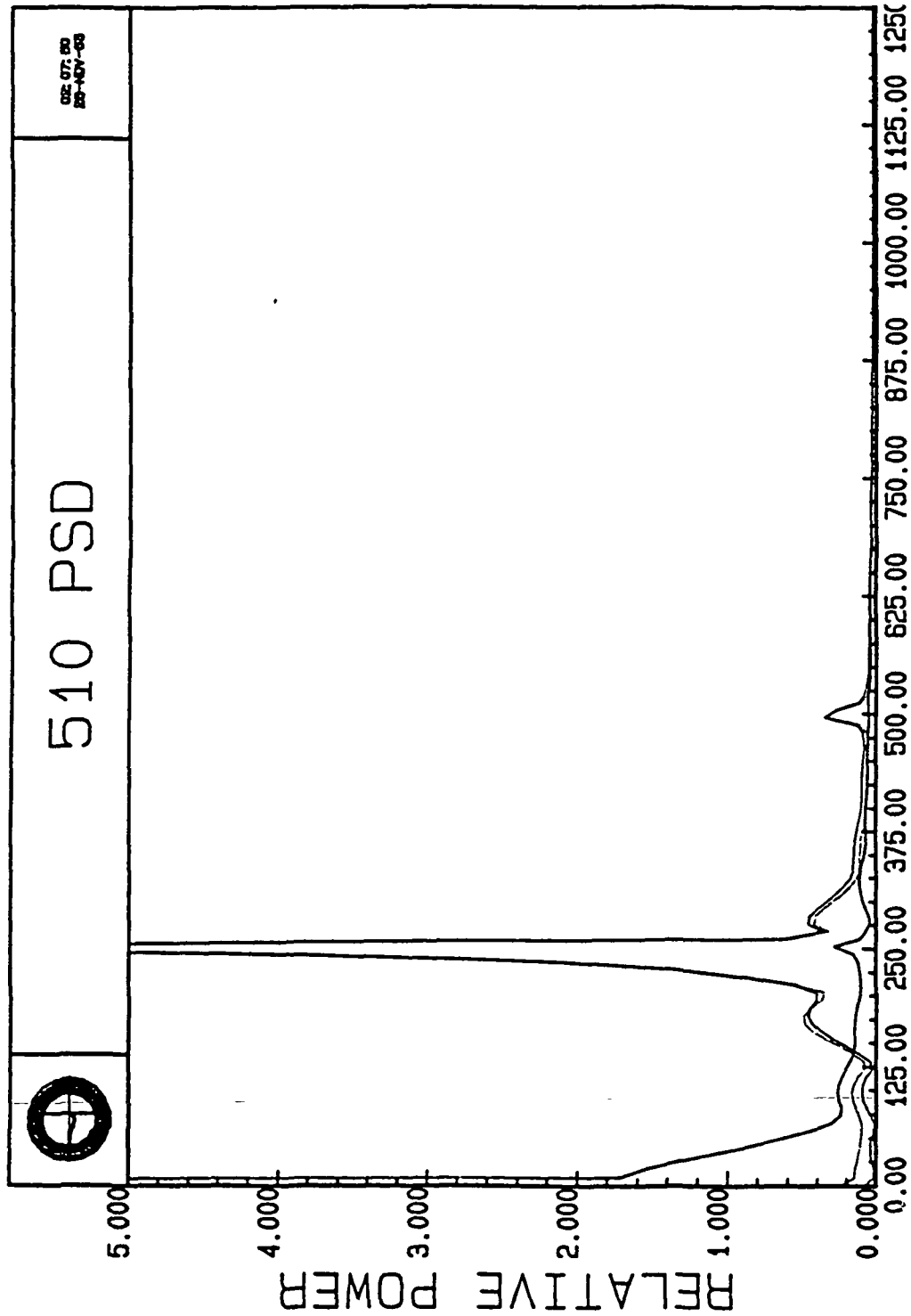

Figure 3.14d PSD of the C-Type Motion; Shaft Speed: 500 Hz.

Let's begin the analysis by examining the vector force diagram presented in Figure 3.15. These vectors are not exactly to scale as far as the indicated magnitudes are concerned. The two parameters defining the equilibrium point are r_o and ϕ_o. r_o is the magnitude of the displacement of the rotor center from the central position with ϕ_o defining the angle made by r_o with the horizontal axis. If the rotor is stationary in the position defined by r_o and ϕ_o, then the forces in the radial and transverse directions must both sum to zero. Because they have been balanced by the side force, force terms arising due to the spin of the shaft have no influence on the equilibrium point. Two equations are written by inspection from Figure 3.15,

$$K_B (r_o - g) + K_S r_o = F_S \cos \phi_o \tag{3.48}$$

$$Q_S r_o = F_S \sin \phi_o \tag{3.49}$$

where F_S is used to denote the side force magnitude. If we define the variable F_r to be the side force acting radially and F_ϕ to be the side force acting tangentially, then the following expression may be used to combine Equations 3.48 and 3.49 and allow for direct solution of r_o.

$$F_S^2 = F_r^2 + F_\phi^2 \tag{3.50}$$

Hence,

$$[(K_B + K_S)r_o - K_B g]^2 + Q_S^2 r_o^2 = F_S^2 \tag{3.51}$$

or,

$$[(K_B + K_S)^2 + Q_S^2]r_o^2 - 2(K_B + K_S)K_B g r_o + [K_B^2 g^2 - F_S^2] = 0 . \tag{3.52}$$

The solutions for r_o are

$$r_o = \frac{(K_B + K_S)K_B g \pm \sqrt{F_S^2[(K_B + K_S)^2 + Q_S^2] - Q_S^2 K_B^2 g^2}}{(K_B + K_S)^2 + Q_S^2} \tag{3.53}$$

Close inspection of the term under the radical indicates that for a zero deadband, any positive value of F_S will yield a positive real solution for r_o. The presence of a deadband imposes another constraint on the conditions used in the evaluation of r_o. If the magnitude of the side force is insufficient to push the rotor out to the deadband, that is, it cannot overcome the seal stiffness forces, then the bearing stiffness plays no role in determining r_o. The minimum side force required to move the rotor out to the deadband can be easily obtained by setting $r_o = g$ and $K_B = 0$ in Equation 3.51. That value of F_S, referred to henceforth, as F_{SMIN} is

$$F_{SMIN} = \sqrt{(K_S^2 + Q_S^2)g^2} . \tag{3.54}$$

For any value of side force less than F_{SMIN}, the value of r_o is the positive

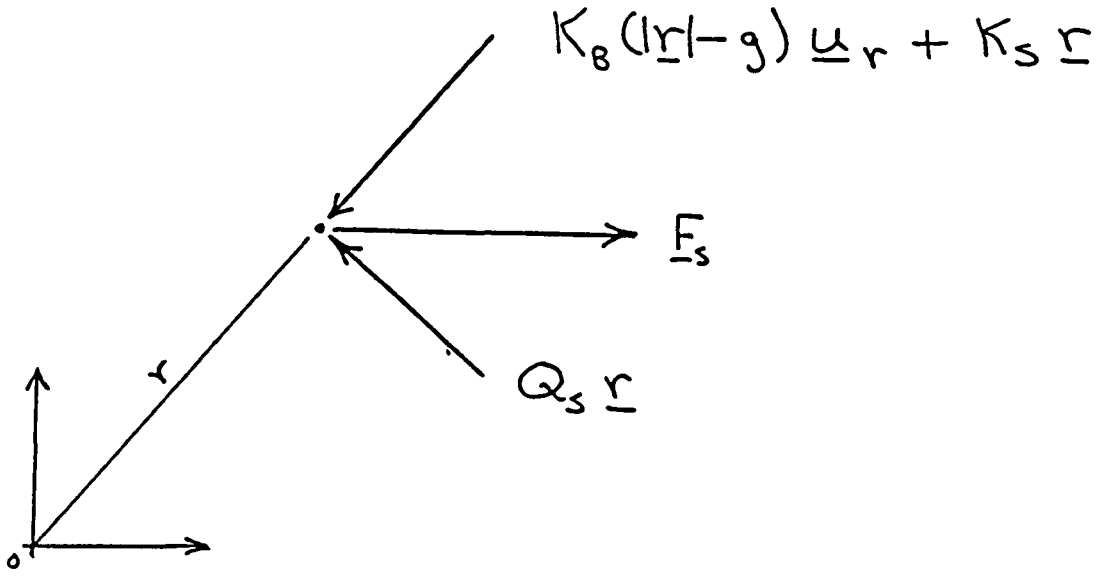

Figure 3.15 Force Diagram Used to Determine the Equilibrium Point.

solution to Equation 3.53 with K_B set to zero. Therefore, any positive value of side force will yield a positive real solution to r_o, provided the conditions outlined above regarding F_{SMIN} are adhered to.

The value of the angle ϕ_o may be acquired most easily from Equation 3.49 once the value of r_o has been computed. The equilibrium point is now completely defined. To confirm that our equations defining the equilibrium point are correct, the simulation is used with the results compared to the values of r_o and ϕ_o computed using Equations 3.53 and 3.49. The comparisons are presented in Table 3.1. The simulation has been executed using four different side force values for each of the five deadband values under consideration. Notice that in some instances, two positive solutions for r_o exist, one located inside the deadband and one outside. The system tends toward the equilibrium point outside the deadband. This response is due to the fact that the side force is greater than F_{SMIN} and, therefore, will displace the rotor to an equilibrium point outside the deadband.

TABLE 3.1
Comparison of Analytical & Numerical Equilibrium Point Values

RUN#	ROOTS TO EQUATION 3.53 r_1 (mils)	r_2 (mils)	ANALYTICAL r_o (mils)	NUMERICAL r_o (mils)	ANALYTICAL ϕ_o(rad)	NUMERICAL ϕ_o(rad)
101	0.862744	-0.082863	0.8672744	0.8672744	0.468705	0.468705
102	1.026741	-0.246860	1.026741	1.026741	0.415011	0.415011
103	1.189615	-0.409734	1.189615	1.189615	0.383025	0.383025
104	1.351936	-0.572055	1.351936	1.351936	0.361776	0.361776
201	1.218376	0.341386	1.218376	1.218376	0.691820	0.691820
202	1.391641	0.168121	1.391641	1.391641	0.578174	0.578174
203	1.559762	0.0	1.559762	1.559762	0.512105	0.512104
204	1.725487	-0.164802	1.725487	1.725487	0.468704	0.468704
301	1.544214	0.795428	1.544214	1.544215	0.941681	0.941680
302	1.737397	0.602245	1.737397	1.737398	0.750868	0.750868
303	1.915547	0.424095	1.915547	1.915547	0.645736	0.645735
304	2.087461	0.252181	2.087461	2.087462	0.578174	0.578174
401	0.788707	-0.164802	*	UNSTABLE CONFIGURATION		
402	2.058953	1.060571	2.058953	2.058952	0.941681	0.941681
403	2.254956	0.864668	2.254856	2.254856	0.787205	0.787205
404	2.436752	0.682771	2.436752	2.436753	0.691820	0.691820
501	0.822743	-0.082867	*	C-TYPE ORBIT		
502	1.066741	-0.246860	*	C-TYPE ORBIT		
503	2.573691	1.325714	2.573691	2.573691	0.941680	0.941684
504	2.771480	1.125925	2.771480	2.771480	0.811864	0.811864

* Indicates no equilibrium point outside the deadband.

For case numbers[1] 401, 501, and 502, no equilibrium points exist outside the deadband because the side force is less than F_{SMIN}. The simulation results for cases 501 and 502 were C-type motions which are subsynchronous. This is generally the type of behavior that is observed when r_o is inside the deadband, in the absence of imbalance forces. Case 401 is an unstable case, a point that

[1] The numbering convention explained in the previous chapter is employed here.

will be explained later. For illustrative purposes, the orbits for cases 501 and 502 are presented in figures 3.16 and 3.17, respectively. Figure 3.18 is a plot of a typical system response as the rotor is displaced to its equilibrium point from the rest position.

With the equilibrium points well in hand, we may now proceed. A slightly different approach is taken in the exploration of the stability properties of the rotor model with a side force acting. Recall the vector equation which describes the system. It is repeated here, with the imbalance term omitted.

$$m\ddot{\underline{r}} = -K_B(r - g)\, u\,(r - g)\,\underline{e}_r - K_S r\underline{e}_r + Q_S\underline{e}_X \times r\underline{e}_r - C_S\dot{\underline{r}} + C_Q\underline{e}_X \times \dot{\underline{r}} \qquad (3.55)$$

We make the following definitions for $\underline{r}$ and $\underline{e}_r$,

$$\underline{r} = \underline{r}_0 + \underline{\delta} \qquad (3.56)$$

$$r = r_0 + \delta_y \qquad (3.57)$$

$$\underline{e}_r = \underline{e}_{r_0} + \delta\underline{e}_r \qquad (3.58)$$

where $\underline{r}_0$ is the equilibrium position vector, $\underline{e}_{r_0}$ is the unit vector in the direction of $\underline{r}_0$, and $\underline{\delta}$ and $\delta\underline{e}r$ are the perturbations associated with $\underline{r}$ and $\underline{e}_r$, respectively. The radial component of $\underline{r}$ with its perturbation is Equation 3.57. Another way to express $\underline{e}_r$ is

$$\underline{e}_r = \frac{\underline{r}_0 + \underline{\delta}}{|\underline{r}_0 + \underline{\delta}|} = \frac{\underline{r}_0}{|\underline{r}_0|} + \frac{\underline{\delta}}{|\underline{r}_0|} - \frac{\underline{r}_0\,\underline{r}_0}{|r_0|^2} \cdot \underline{\delta} + \ldots \qquad (3.59)$$

which may also be expressed as:

$$\underline{e}_r = \underline{e}_{r_0} + \frac{\delta_z}{r_0}\,\underline{e}_\phi \quad . \qquad (3.60)$$

We now examine the nonlinear deadband term in equation 3.55. In the small and to a first order approximation,

$$-K_B\,(r - g)\,u\,(r - g)\,\underline{e}_r = -K_B\,(r_0 - g + \delta_y)\left(\underline{e}_{r_0} + \frac{\delta_z}{r_0}\,\underline{e}_\phi\right) \qquad (3.61)$$

$$= -K_B r_0\underline{e}_{r_0} + K_B g\underline{e}_{r_0} - K_B\delta_y\underline{e}_{r_0} - K_B\,\frac{(r_0 - g)}{r_0}\,\delta_z\underline{e}_\phi$$

with δ_y and δ_z having the same definition as in previous sections. It follows, therefore, that in terms of the perturbation variables, the system may be expressed in the following form:

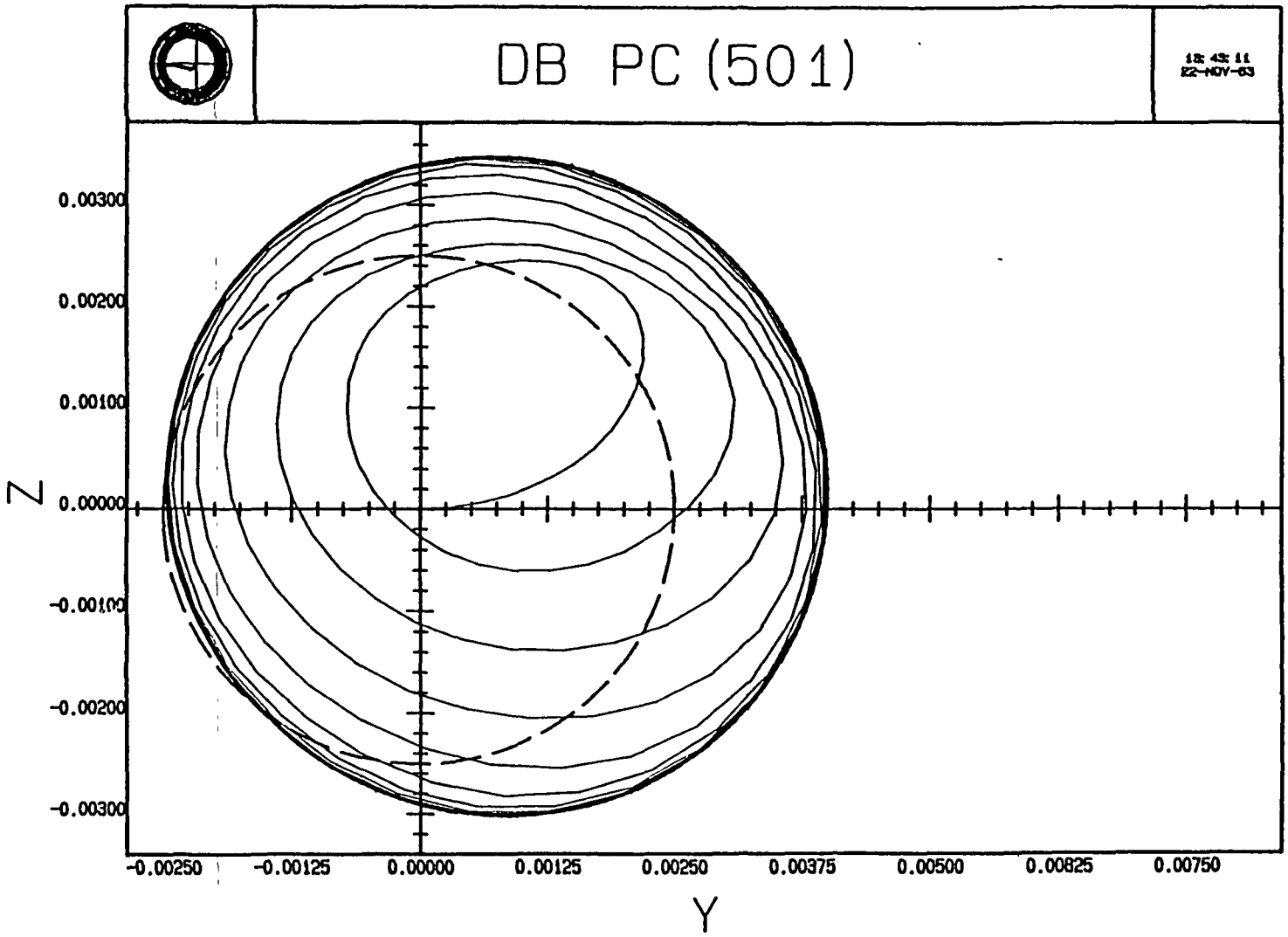

Figure 3.16a Subsynchronous C-Type Whirl Resulting from no Equilibrium Point.

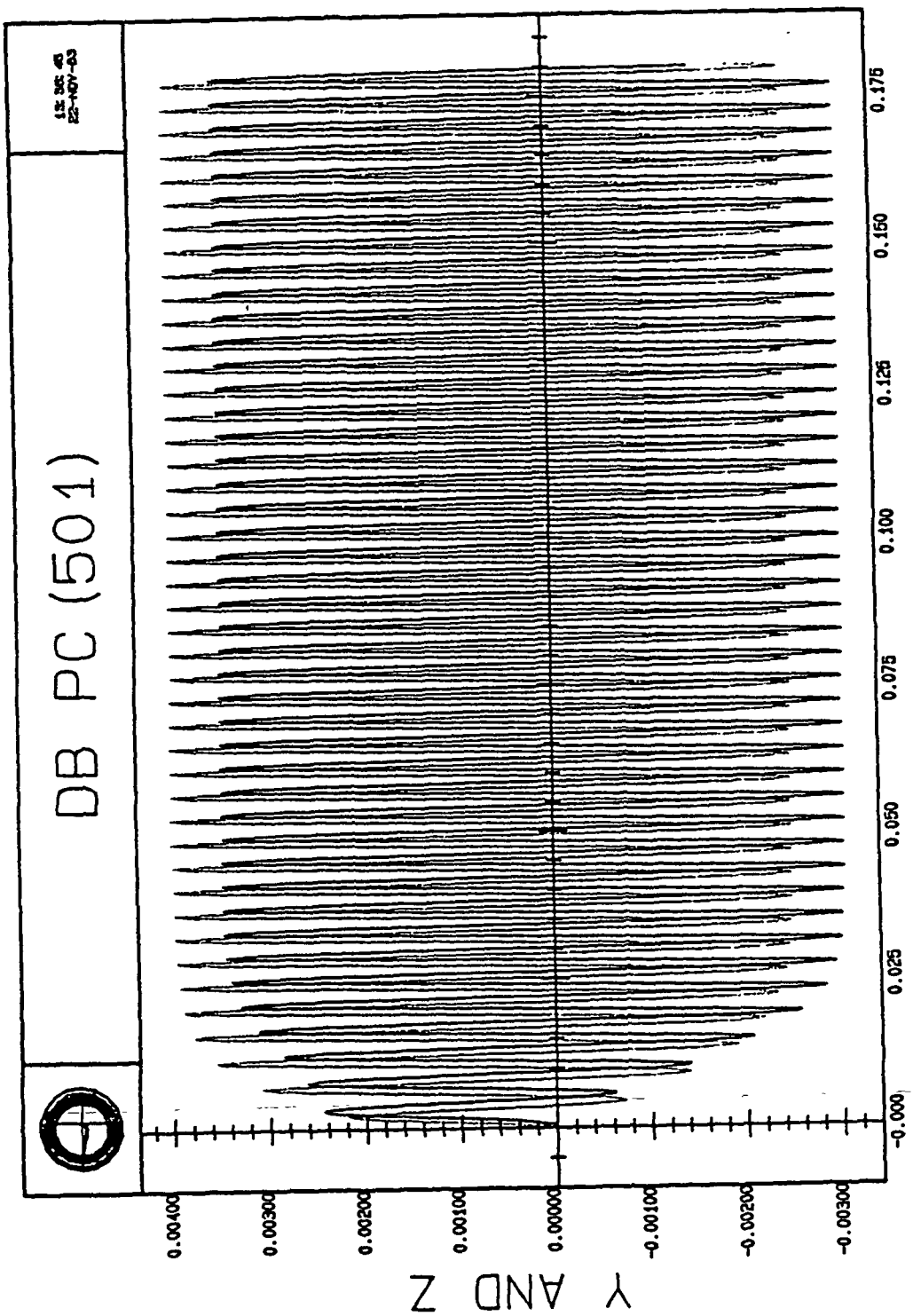

Figure 3.16b The x and y Components as a Function of Time.

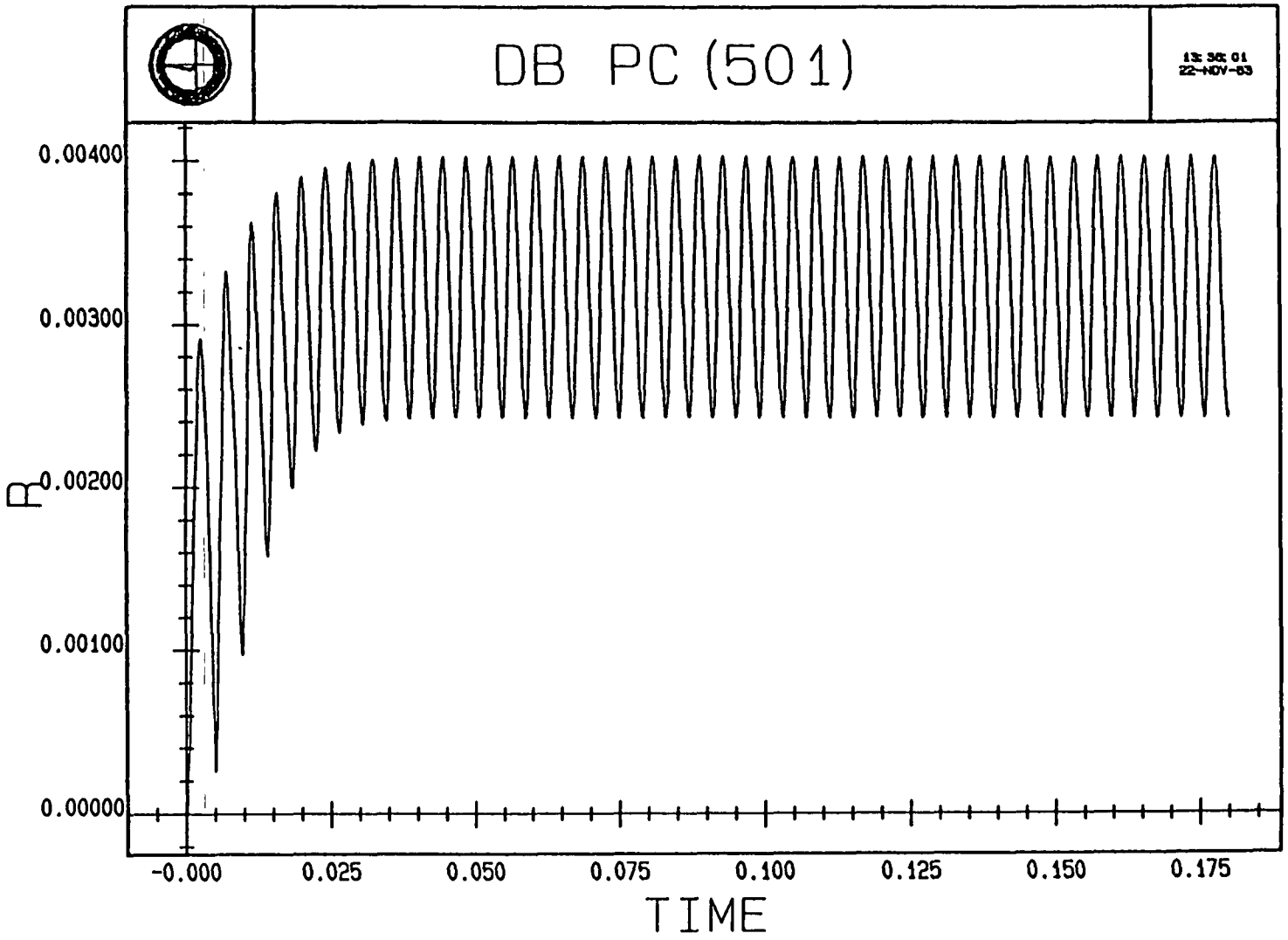

Figure 3.16c The Orbit Radius as a Function of Time.

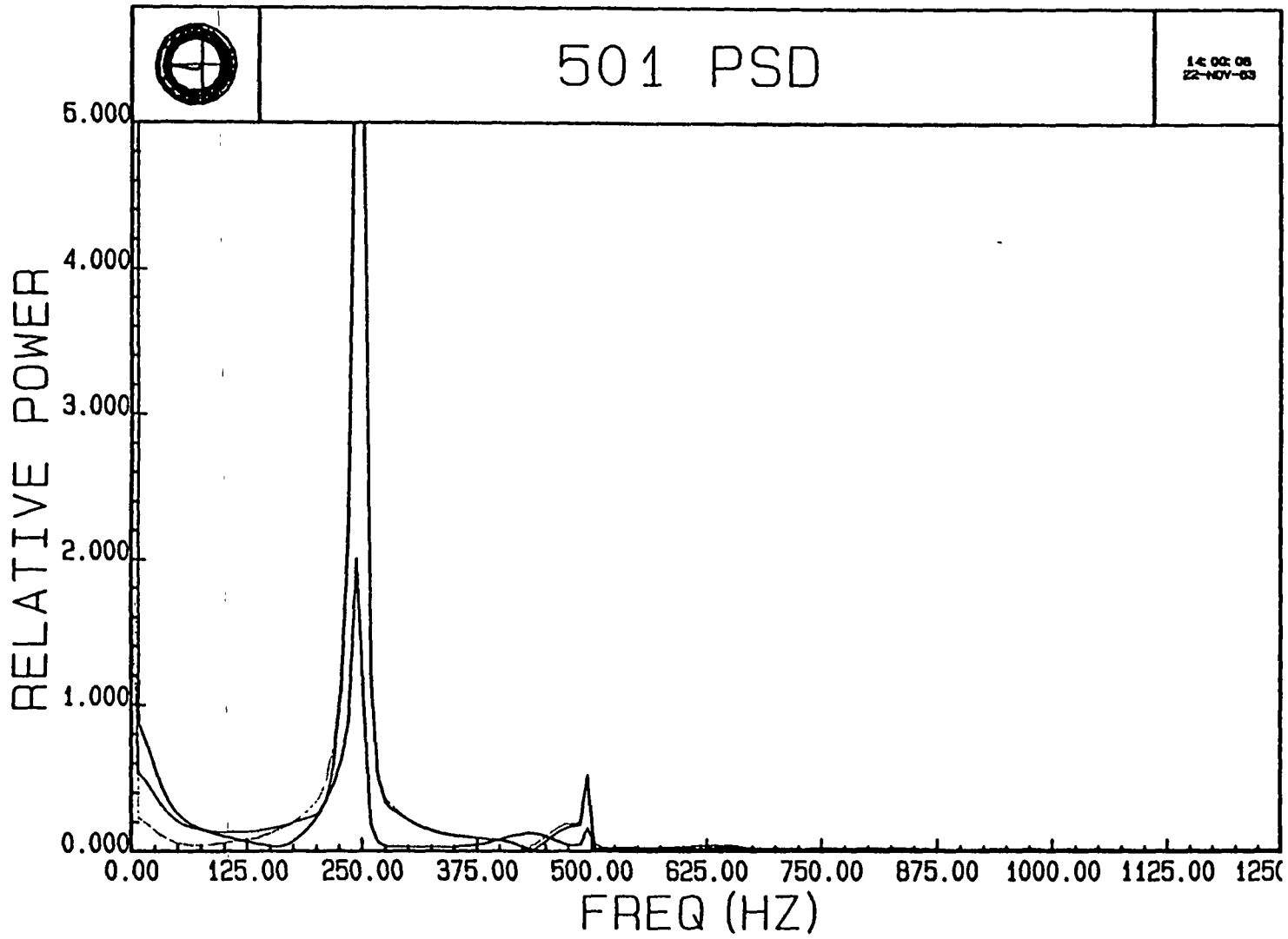

Figure 3.16d PSD of the C-Type Motion; Shaft Speed: 500 Hz.

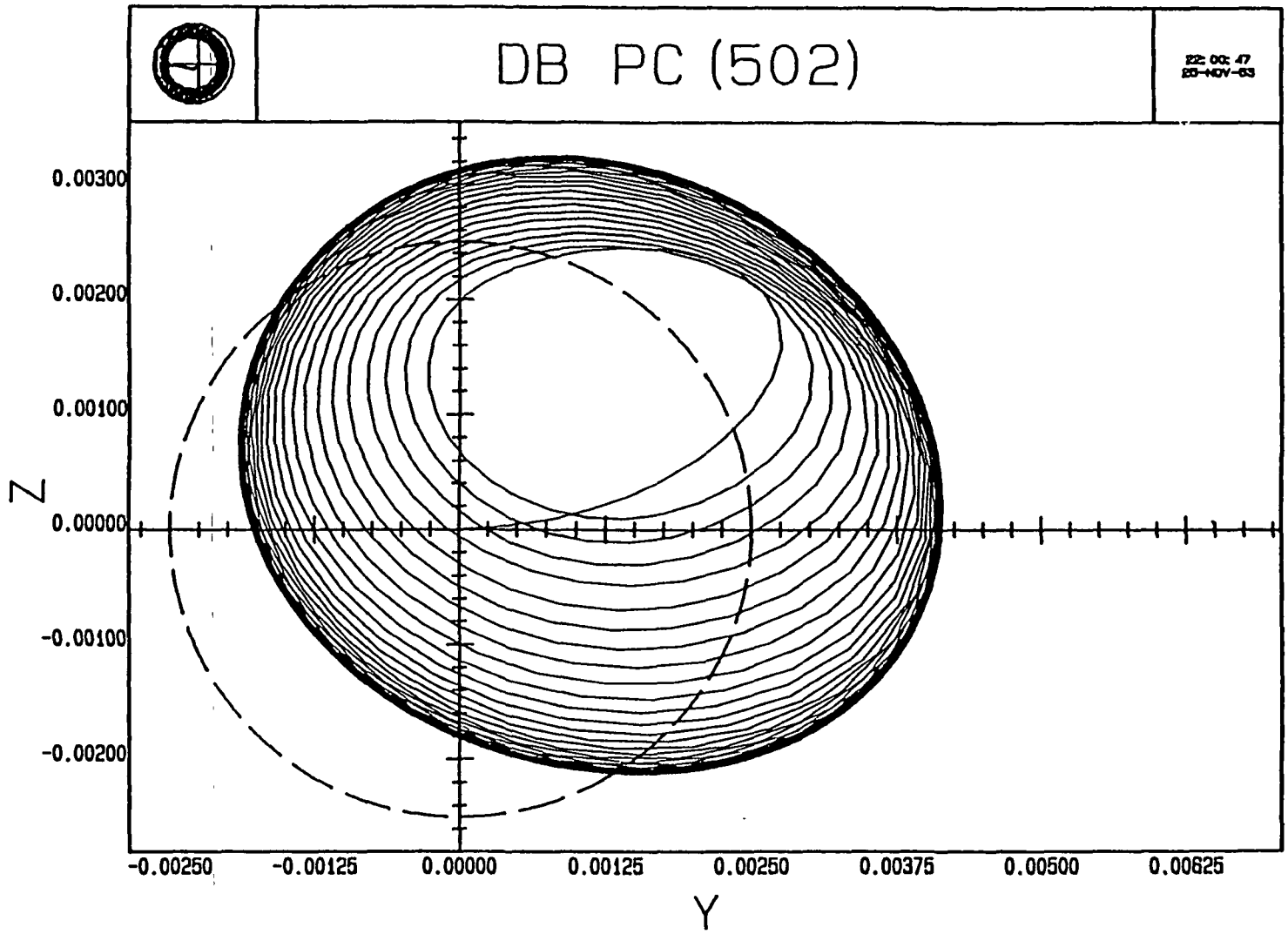

Figure 3.17a Subsynchronous C-Type Whirl Resulting From No Equilibrium Point.

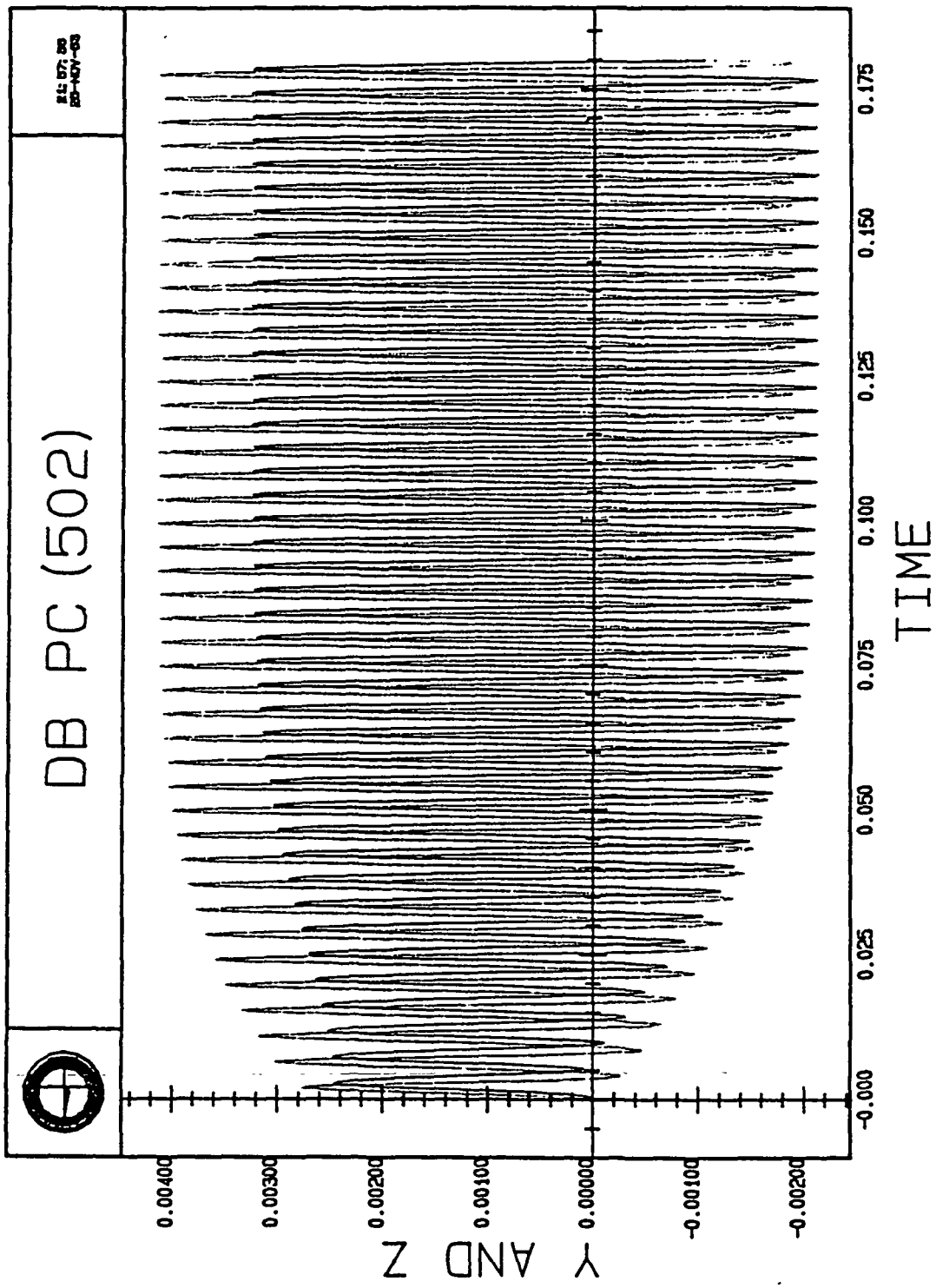

Figure 3.17b The x and y Components as a Function of Time.

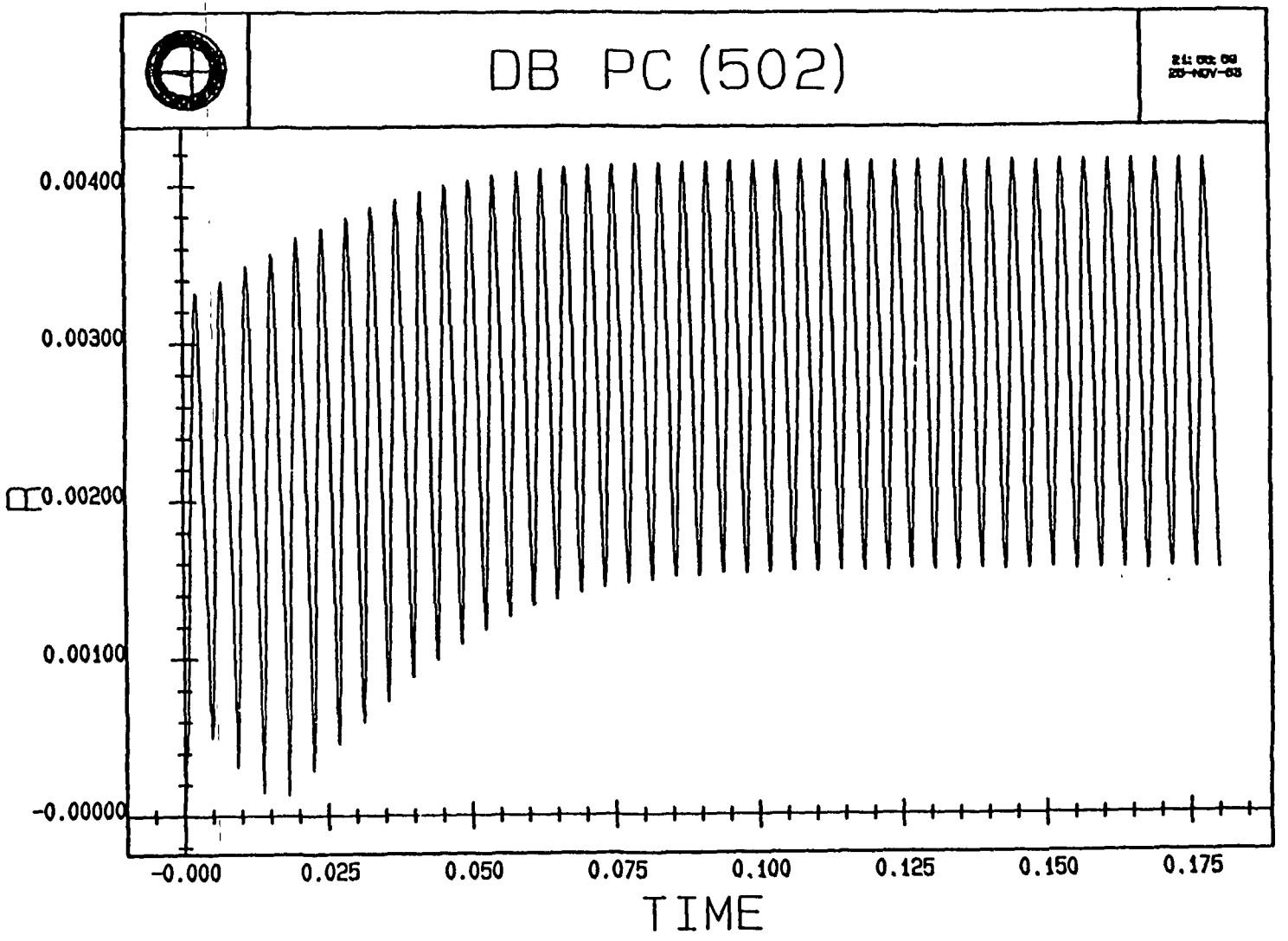

Figure 3.17c The Orbit Radius as a Function of Time.

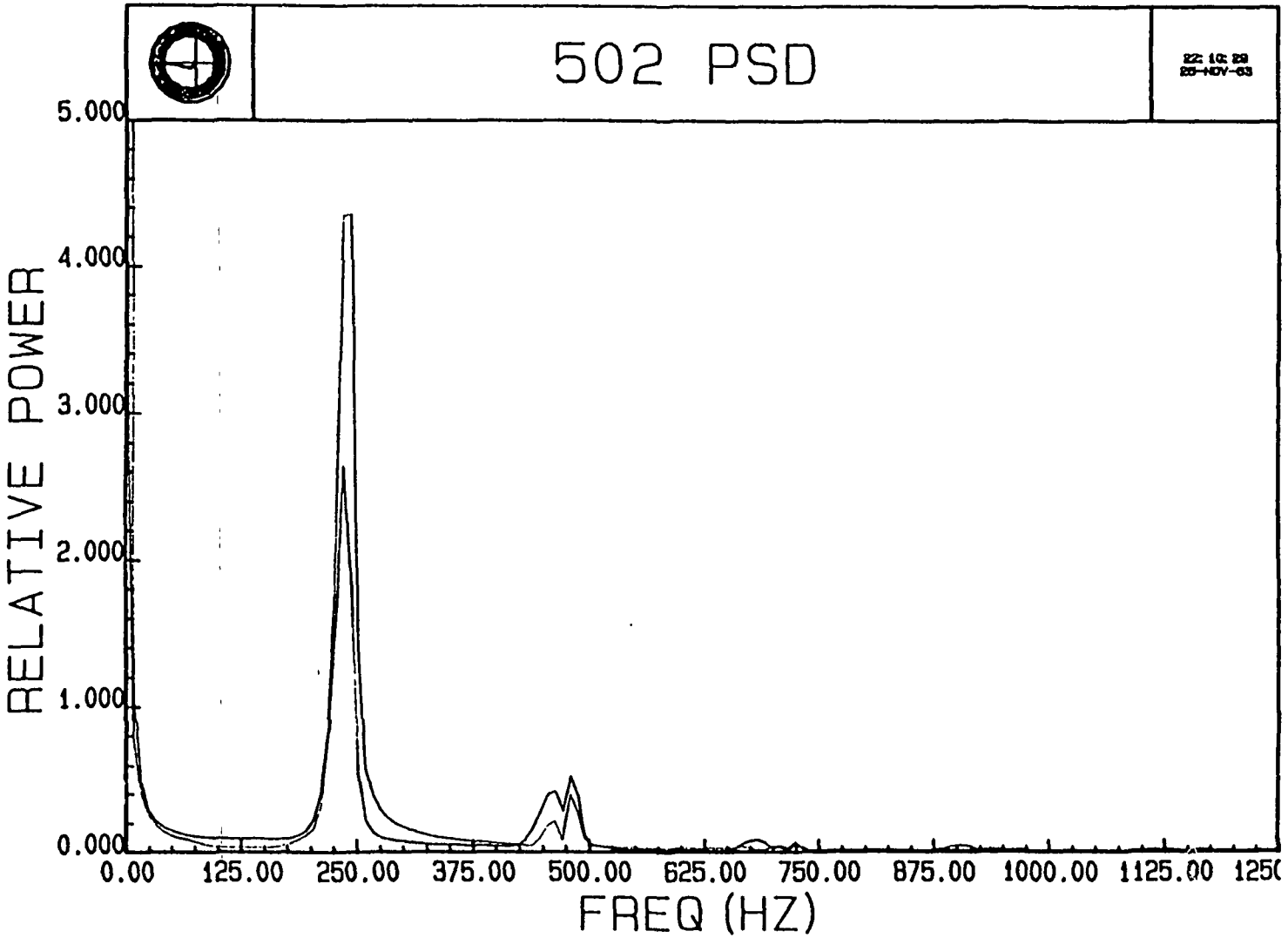

Figure 3.17d PSD of the C-Type Motion; Shaft Speed: 500 Hz.

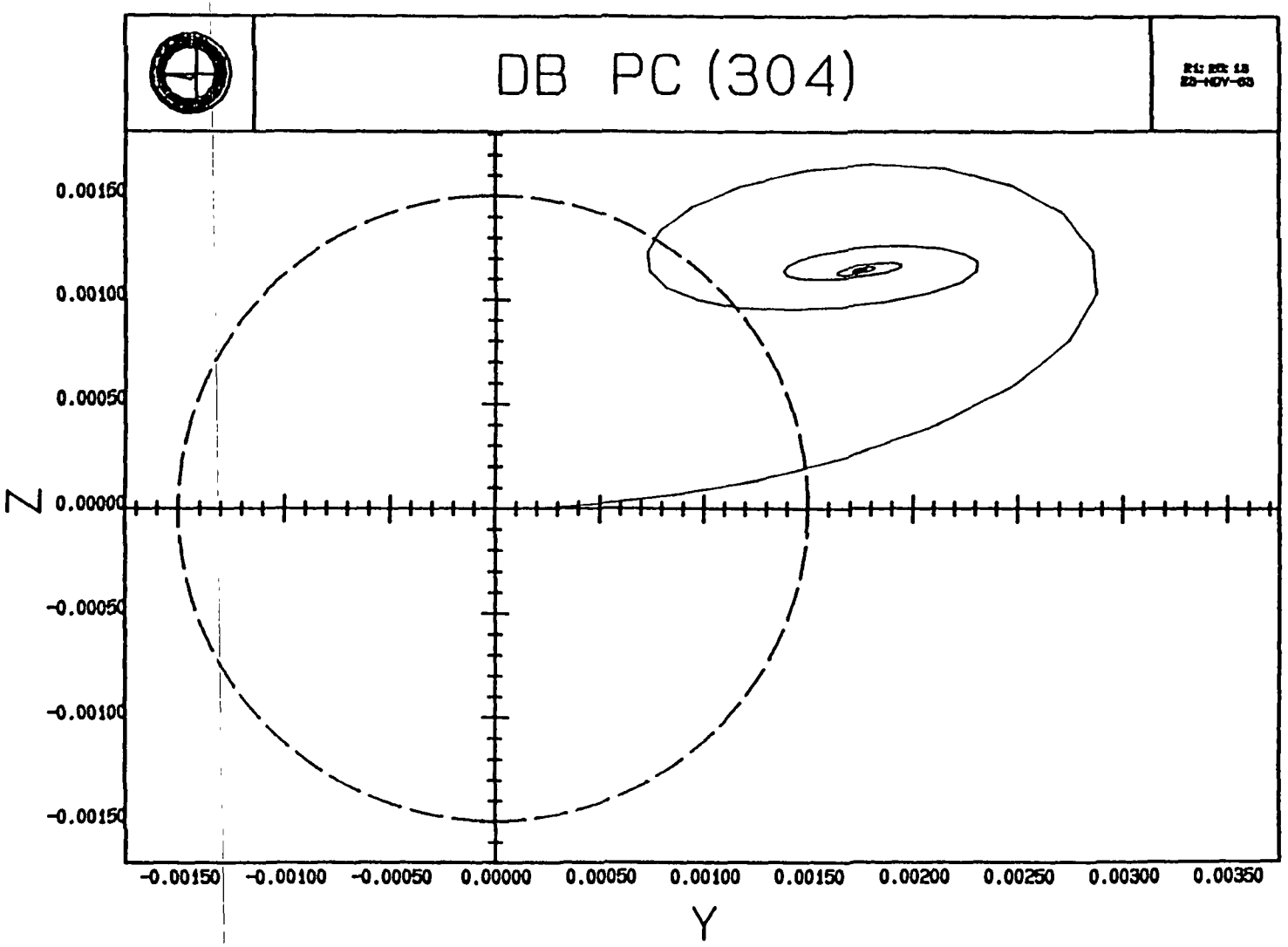

Figure 3.18a Typical Equilibrium Point Response.

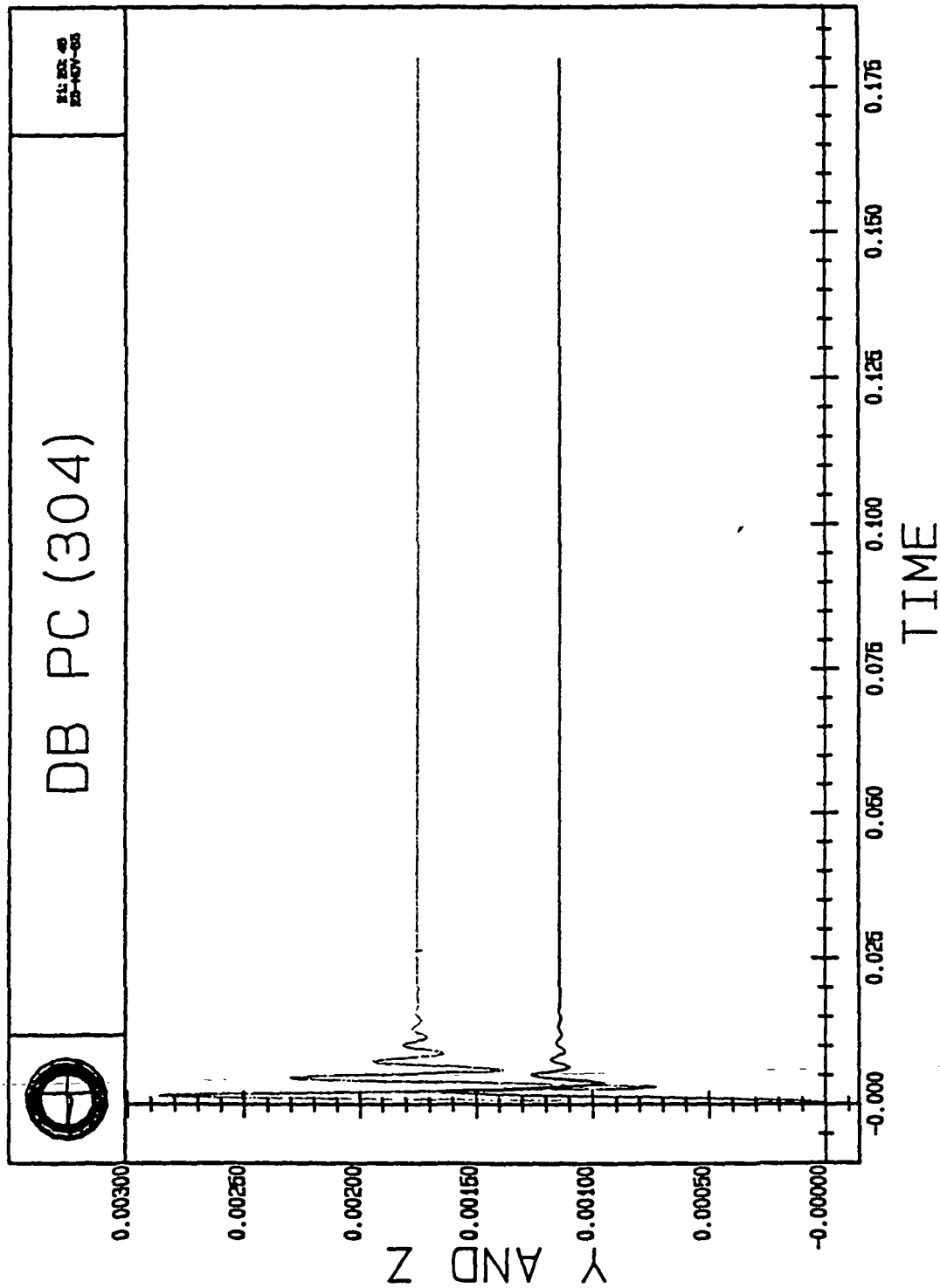

Figure 3.18b The x and y Components as a Function of Time.

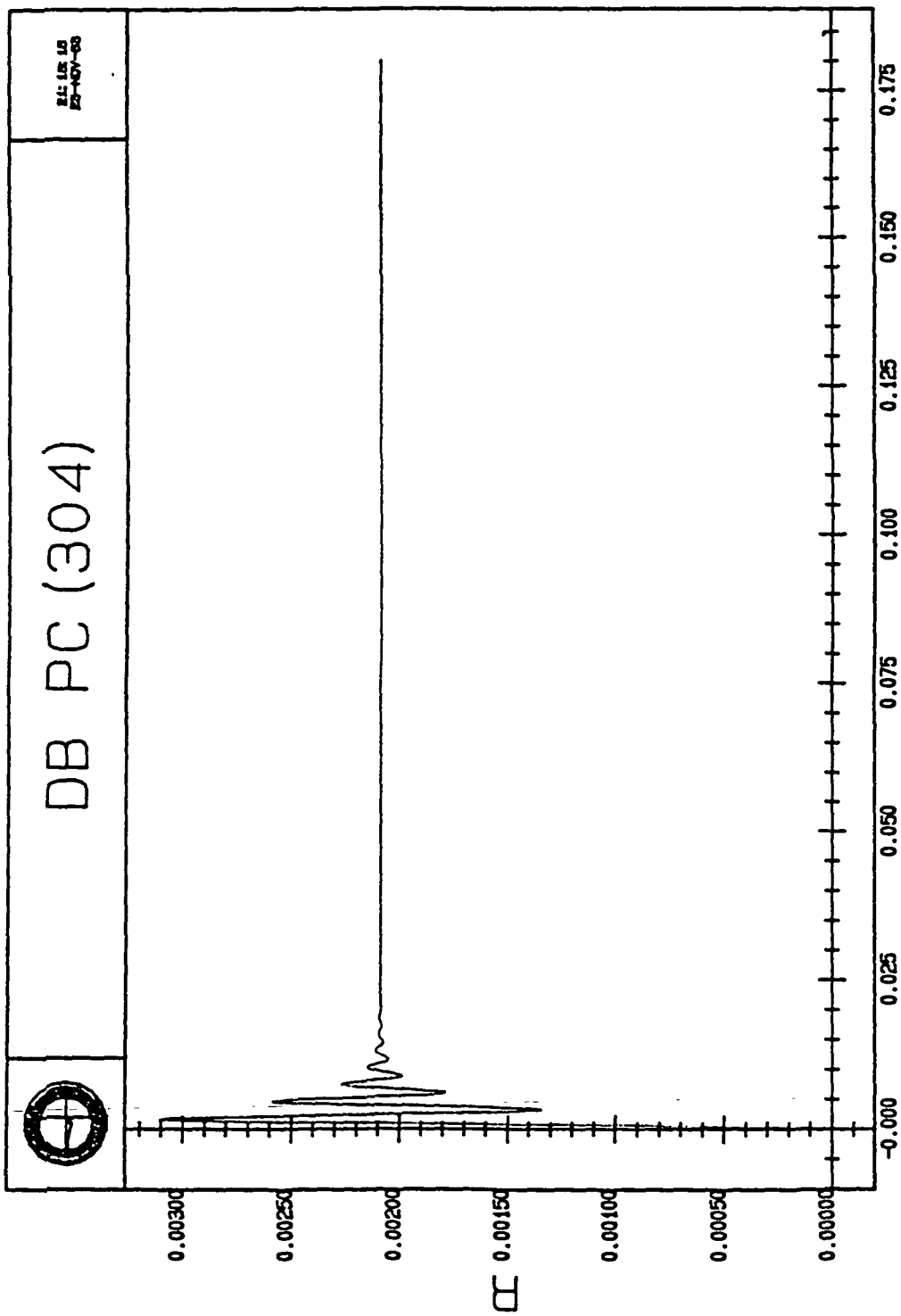

Figure 3.18c The Orbit Radius as a Function of Time.

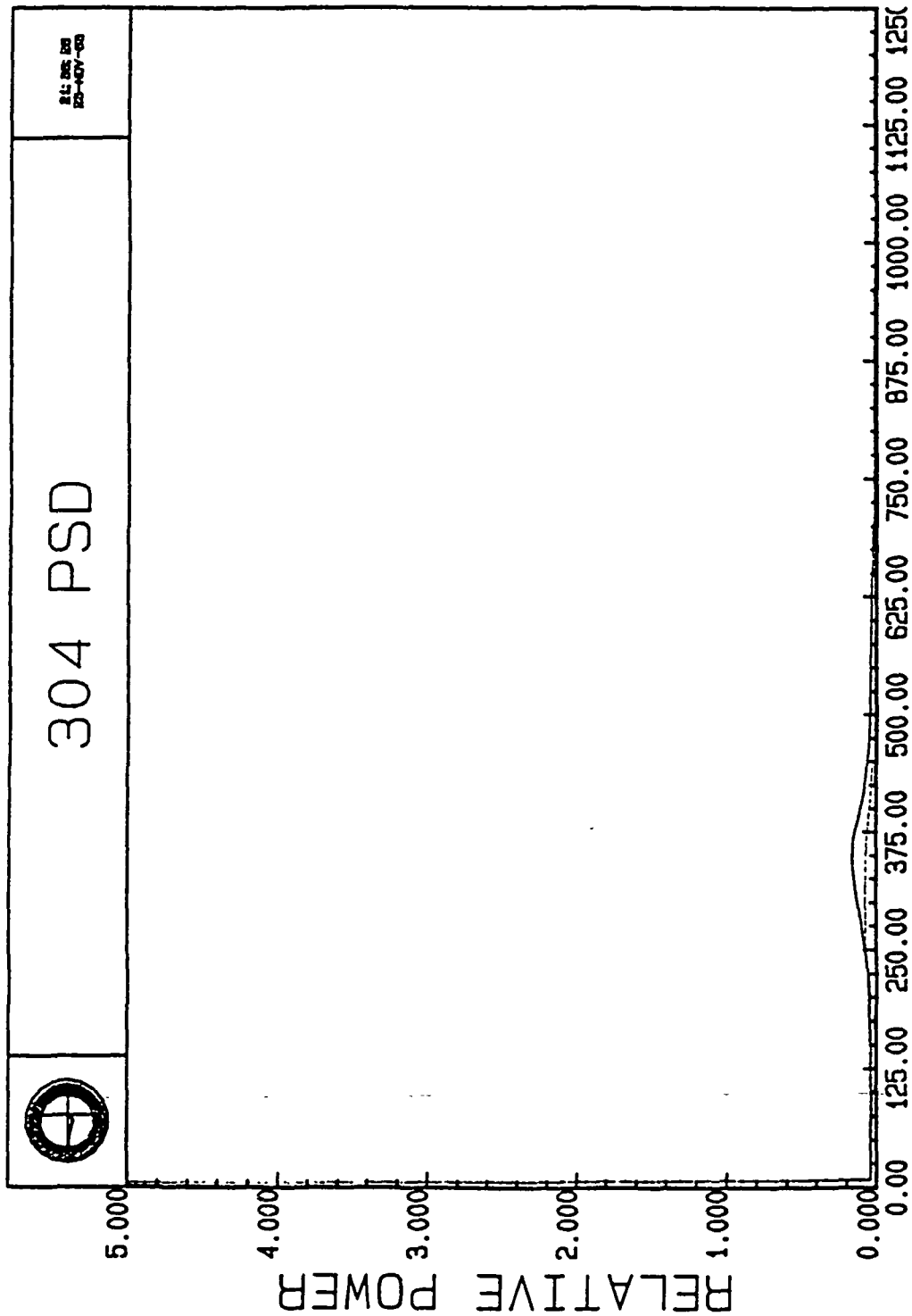

Figure 3.18d PSD of the C-Type Motion; Shaft Speed: 500 Hz.

$$m \begin{bmatrix} \ddot{\delta}_y \\ \ddot{\delta}_z \end{bmatrix} = - \begin{bmatrix} K_S + K_B & Q \\ -Q & K_S + K_B (1 - g_o) \end{bmatrix} \begin{bmatrix} \delta_y \\ \delta_z \end{bmatrix}$$

$$- \begin{bmatrix} C_S & C_Q \\ -C_S & C_S \end{bmatrix} \begin{bmatrix} \dot{\delta}_y \\ \dot{\delta}_z \end{bmatrix} . \qquad (3.62)$$

Where $g_o = \frac{g}{r_o}$.

The effects of side forces are now inherent in the formulation. Stability may be assessed through examination of equation 3.62. State assignments for the perturbation variables are given below.

$$x_1 = \delta_y \qquad x_3 = \delta_z$$

$$x_2 = \dot{\delta}_y \qquad x_4 = \dot{\delta}_z \qquad (3.63)$$

Rewriting 3.62 in state varible format yields the following differential equations.

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = - \frac{K}{m} x_1 - \frac{C_S}{m} x_2 - \frac{Q_S}{m} x_3 - \frac{C_Q}{m} x_4$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{Q_S}{m} x_1 + \frac{C_Q}{m} x_2 - \frac{K_S + K_B (1 - g_o)}{m} x_3 - \frac{C_S}{m} x_4 \qquad (3.64)$$

The sum $K_B + K_S$ has been replaced by K for simplicity's sake. The system matrix is formed as before from which the characteristic equation is derived by solving the determinant of $[sI - A]$.

$$\det [sI - A] = \begin{vmatrix} s & -1 & 0 & 0 \\ \frac{K}{m} & s + \frac{C_S}{m} & \frac{Q_S}{m} & \frac{C_Q}{m} \\ 0 & 0 & s & 1 \\ \frac{-Q_S}{m} & \frac{-C_Q}{m} & \frac{K_S+K_B\ (1\ -\ g_0)}{m} & s + \frac{C_S}{m} \end{vmatrix} \tag{3.65}$$

The characteristic equation, P(s) is

$$P(s) = s^4 + \frac{2C_S}{m} s^3 + \frac{C_S^2 + K_m}{m^2} s^2 + \frac{C_Q^2 + m[K_S+K_B(1- g_0)]+KC_S+mC_QQ_S}{m^2} s$$

$$+ [\frac{C_QQ_S + Q_S^2 + [K_S + K_B(1 - g_0)]\ C_S + Q_S^2 + K\ [K_S + K_B(1 - g_0)}{m^2}] \tag{3.66}$$

The stability properties of the system may now be determined by examining the roots of equation 3.66. The goal is to establish the stability boundary for the system as a function of side force for various deadbands. That is, to determine the frequency at which the system goes unstable as the side force is varied for a given deadband. The initial attempts to establish the stability boundaries were not successful due to an anomoly in the method used. The idea was to examine the Routh-Herwitz criteria. However, because this method is an iterative procedure, numerical errors due to truncation and roundoff which are made early in the process of generating the array tend to propogate and become greatly amplified rather quickly. Misleading results are obtained. It turns out that attempts to root the polynomial using numerical methods lead to erroneous results as well. Because the roots of the equations are complex with the values of the imaginary parts quite near equal, computational difficulties arise when standard rooting algorithms are used. Therefore, an analytical approach is used.

Inspection of the form of the system matrix reveals that it may be rewritten in the form presented here. The term C_Q has been omitted to simplify the analysis. This is justified because of the relative magnitude of this parameter; it is small compared to the others. Its ommission has no apparent effect on the analysis.

$$P(s) = \det \begin{bmatrix} s^2 + \frac{C_S}{m} s + \frac{K_S + K_B}{m} & \frac{Q_S}{m} \\ \frac{-Q_S}{m} & s^2 + \frac{C_S}{m} s + \frac{K_S + K_B\,(1 - g_o)}{m} \end{bmatrix} \tag{3.67}$$

We make the following definitions:

$$\frac{K_S + K_B}{m} = K + \Delta \tag{3.68}$$

$$\frac{K_S + K_B\,(1- g_o)}{m} = K - \Delta. \tag{3.69}$$

The solutions for K and Δ are easily determined to be

$$K = \frac{K_S + K_B}{m} - \frac{K_B g}{2mr_o} \tag{3.70}$$

$$\Delta = \frac{K_B g}{2mr_o}. \tag{3.71}$$

Using the expressions in equations 3.68 and 3.69 the characteristic equation determinant becomes

$$\begin{vmatrix} s^2 + \frac{C_S}{m} s + K + \Delta & \frac{Q}{m} \\ \frac{-Q}{m} & s^2 + \frac{C_S}{m} s + K - \Delta \end{vmatrix} \tag{3.72}$$

with the resulting characteristic equation

$$P(s) = \left[s^2 + \frac{C_s}{m} s + K + \Delta \right] \left[s^2 + \frac{C_s}{m} s + K - \Delta \right] + \frac{Q^2}{m^2} \qquad (3.73)$$

or

$$P(s) = \left[s^2 + \frac{C_s}{m} s + K + \Delta^2 - \frac{Q_s^2}{m^2} \right] \left[s^2 + \frac{C_s}{m} s + K - \Delta^2 - \frac{Q_s^2}{m^2} \right] \qquad (3.74)$$

We now have the characteristic equation neatly expressed as the product of two quadratics in s. Extracting the roots is now a simple matter. The expressions are readily rooted and the stability boundaries determined via numerical methods. A computer program has been generated whose function is to compute the equilibrium position as a function of the deadband and side force. This value is then used with the other parameters to determine the roots of the characteristic equation. The real parts of the roots are examined. The value of shaft spin rate that first produces a positive real part of a root is declared to be the stability boundary. This process is iterated for values of side force up to 4000 pounds for each deadband considered. The deadbands used are 0.0, 0.5, 1.0, 1.5, 2.0, and 2.5 mils.

The stability boundaries established are plotted in figure 3.19. Several interesting facts are observed when this figure is examined. The stability boundary for a deadband of zero is a constant 4848 radians/second, the dashed curve in Figure 3.19. This is the same stability boundary established for the simple form of the system examined in section 3.1. This frequency is considered to be the global stability boundary. That is, the system is globally unstable when run at frequencies higher than this value. Local stability, or stability in the small, may be ehnanced by other factors.

Let's take a moment to examine what's going on within the system at various points along the stability boundaries. For our system, when the side force magnitude is insufficient to push the rotor out to the deadband, i.e. the side force is less than F_{SMIN}, then the stability boundary is constant at 1979.2 radians per second. This frequency is apparently independent of the size of the deadband. What does depend on the deadband is the side force value required to get the rotor out to the deadband. The larger deadbands require larger side forces.

At that point, the stability boundary begins to increase significantly to a maximum. Along this portion of the curve, the instability frequency is that one at which the applied side force is just equal to F_{SMIN}, and the equilibrium point, r_o, is equal to the magnitude of the deadband. The maximum frequency of instability is the one where the side force is of sufficient magnitude to just push the rotor center slightly outside of the deadband area. This frequency is approximately 6200 radians/second. At this point, the instability curve begins a decay, and asymptotically approaches the global stability boundary. On this portion of the curve, the values of side force are such that the equilibrium point is always outside the deadband.

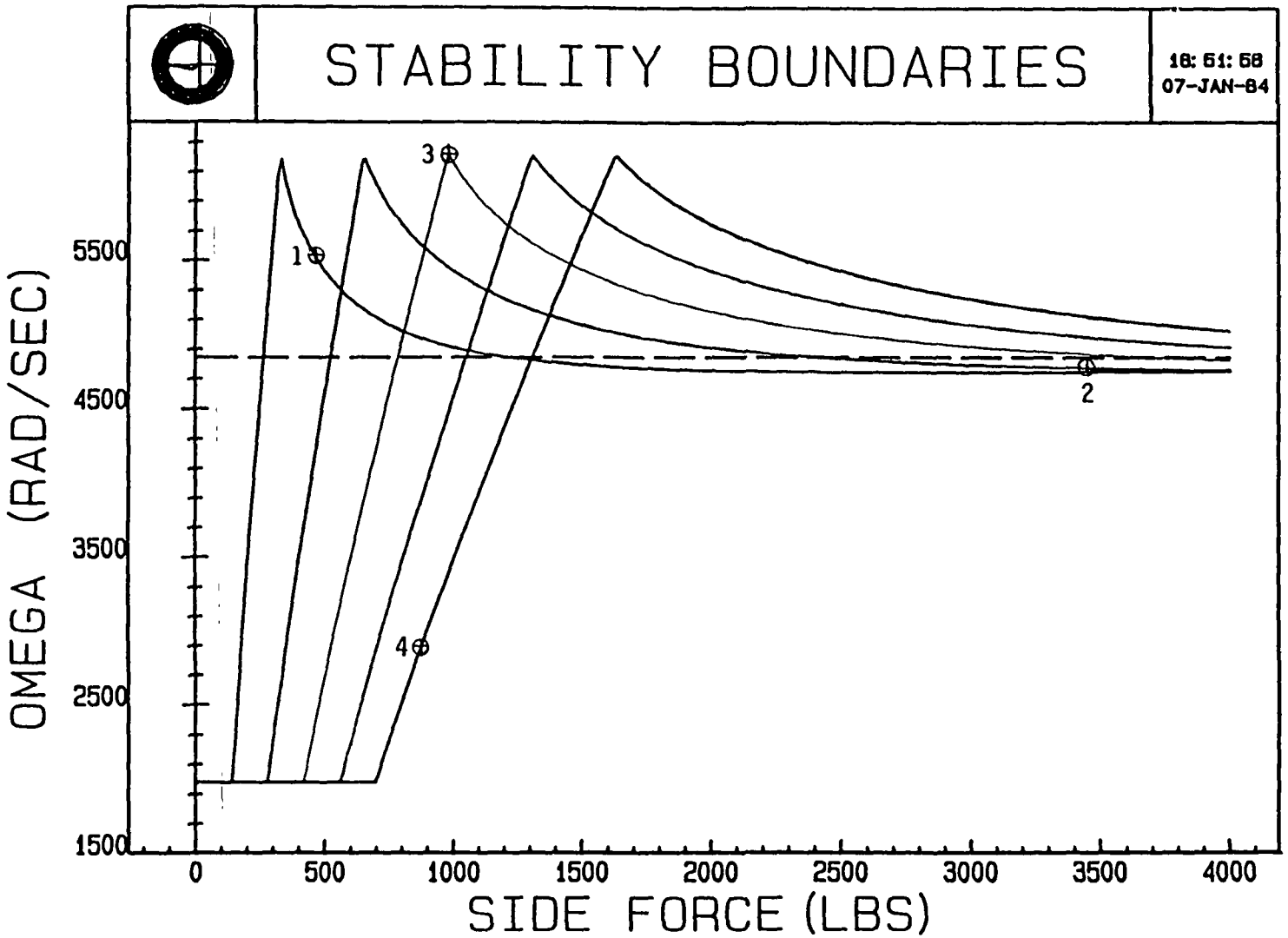

Figure 3.19 Stability Boundaries for the Zero and the Five Non-Zero Deadbands.

Recall that case 401 was referred to in Table 3.1 as being unstable. This case is executed at a frequency of 3141.59 radians/second with a side force of 600 pounds, and a deadband of 2.0 mils. Examination of the stability boundary plot for this value of deadband clearly indicates that this case is outside the boundary.

Notice, in Figure 3.19, that all five of the non-zero deadband stability curves are very similar in their general character. The maxima appear at approximately the same spin frequency as do their minima. We have shown that these curves do, indeed, collapse into a single curve when the system is nondimensionalized. To accomplish this, the units of displacement, force, and time are modified in such a way that the system parameters become unitless. The following table summarizes the way in which the various units are expressed.

Table 3.2
NONDIMENSIONALIZATION

STANDARD SYSTEM	UNIT	NONDIMENSIONALIZED SYSTEM	NEW UNIT DESCIPTION
inches	displacement	g	deadband
pounds	force	$K_S g$	seal stiffness x g
seconds	time	$\sqrt{\frac{m}{K_S + K_B}}$	system natural frequency

Figure 3.20 is a plot of the stability boundary for the normalized system. With this curve and the given conversion factors, one may determine the stability boundary for any deadband value.

Validation and sensitivity studies have been conducted on the established stability boundaries. Four test points were chosen in different positions along the curves. These test points are marked on Figure 3.19. The points were chosen in such a way that each portion of the curves may be examined.

Test point one is selected on the 0.5 mil deadband curve on the portion which is decaying from the maximum. The frequency considered is 5536 radians/second (881.1 Hz) and the side force value is 460 pounds. Test point two is chosen on the 1.0 mil deadband curve in the segment where the curve is near the global stability boundary. The spin frequency at this point is 4774.6 radians/second (760 Hz) with a side force of 3450 pounds. The third test point is picked at a peak value, that of the 1.5 mil deadband curve. The shaft spin rate is 6219.7 radians/second (990 Hz) with a side force value of 980 pounds. The fourth and final test point is located on the 2.5 mil deadband boundary on that part of the curve where the side force just equals F_{SMIN} and the equilibrium point is on the deadband. The frequency at this point is 2896.6 radians/second (461 Hz) with a side force of 880 pounds acting.

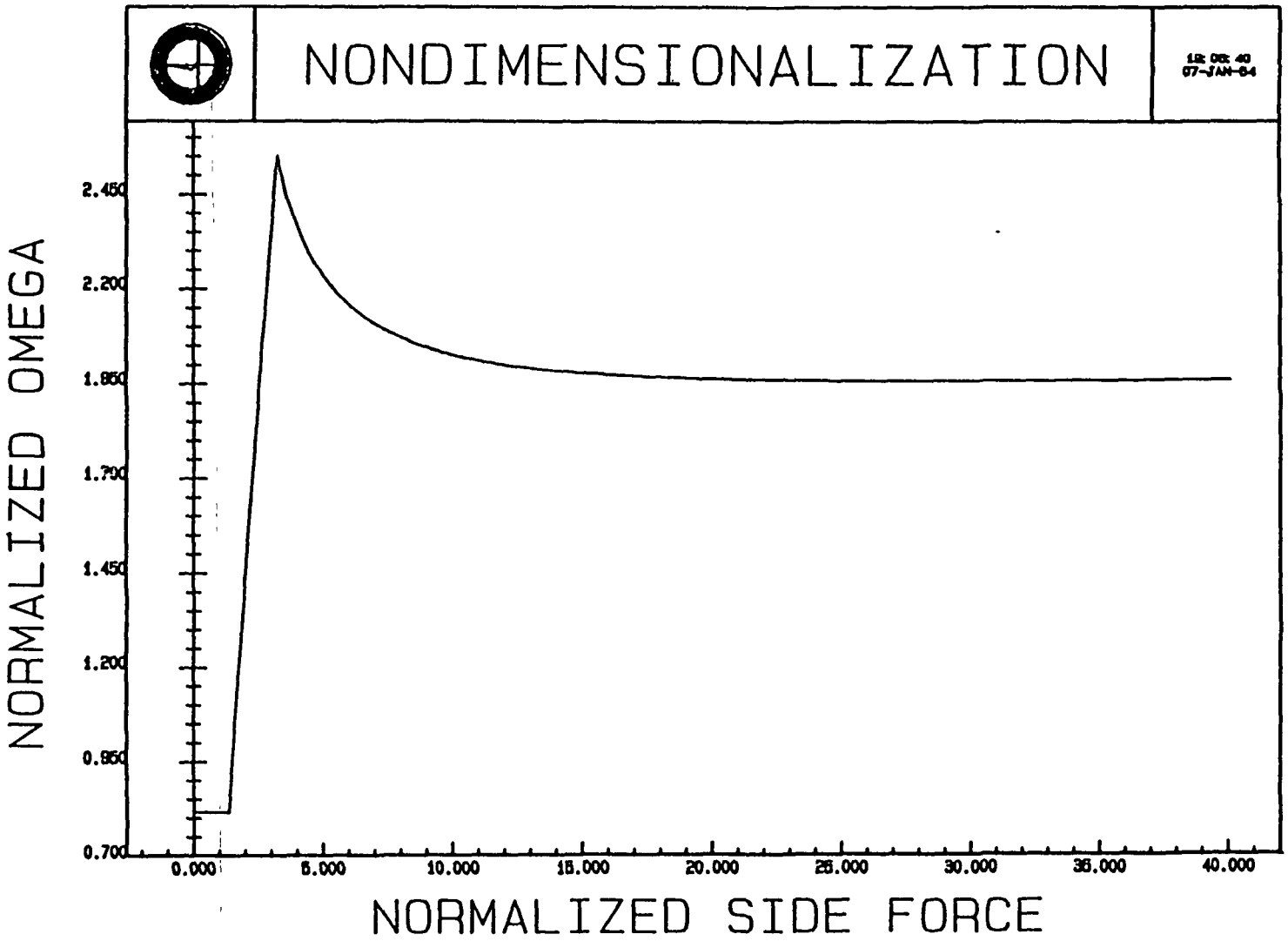

Figure 3.20 Nondimensionalized Stability Boundary.

Verification of the boundaries is performed by executing the simulation with the initial conditions at the four points described being the equilibrium points with no additional forces acting, other that those defined above. In all four cases, the result of the program execution is that the rotor center remains at the equilibrium point and is stable.

To determine their sensitivity to disturbances, two experiments are performed at the test points. First, the shaft spin frequency is increased, leaving all other parameters unchanged, until the system goes unstable. Second, imbalance is introduced, leaving the frequency unchanged. Tabulated below in table 3.3 are the summarized results of the study.

TABLE 3.3. Stability Boundary Sensitivity Study

	TEST PONT NUMBER			
SENSITIVE PARAMETER	1	2	3	
Equilibrium Point	Stable	Stable	Stable	Stable
1% ↑ Shaft Speed	Unstable	Unstable	Very Unstable	Stable
3% ↑ Shaft Speed	Unstable	Unstable	Very Unstable	Stable
5% ↑ Shaft Speed	Very Unstable	Very Unstable	Very Unstable	Unstable
ϵ = 0.01 mils	Stable	Stable	Very Unstable	Stable
ϵ = 0.05 mils	Unstable	Stable	Very Unstable	Stable
ϵ = 0.10 mils	Unstable	Stable	Very Unstable	Stable
ϵ = 0.20 mils	Unstable	Stable	Very Unstable	Stable
ϵ = g	Unstable	Unstable	Very Unstable	Stable

The point most sensitive to any disturbance is point three at the peak of the boundary. One may think of the system stability as a ball perched atop a steep hill, it's right easy to roll off. As indicated in Table 3.3, the slightest disturbance at this point results in unstable behavior. The point least sensitive to disturbances is number four. I haven't come up with a suitable analogy for this one. An imbalance equivalent to the magnitude of the deadband does not produce an instability. A frequency increase of 5% is required to elicit an unstable response. Points one and two proved to quite sensitive to frequency increases, while not quite as sensitive to the addition of rotor eccentricity. For all cases, if the initial condition vector is set such tht any non-zero value of $\underline{\dot{r}}$ or $\dot{\phi}$ is input, the system is immediately unstable.

Several interesting orbits resulted from the verification study. Plotted in Figure 3.21 is an orbit characteristic of the ones obtained when an imbalance is added to test point two. Figure 3.22 is a plot of an unstable orbit caused by the 5% increase in frequency for point four. Finally, Figure 3.23 shows the strange orbit obtained by setting $\epsilon = g$ for test point four. This is a stable orbit.

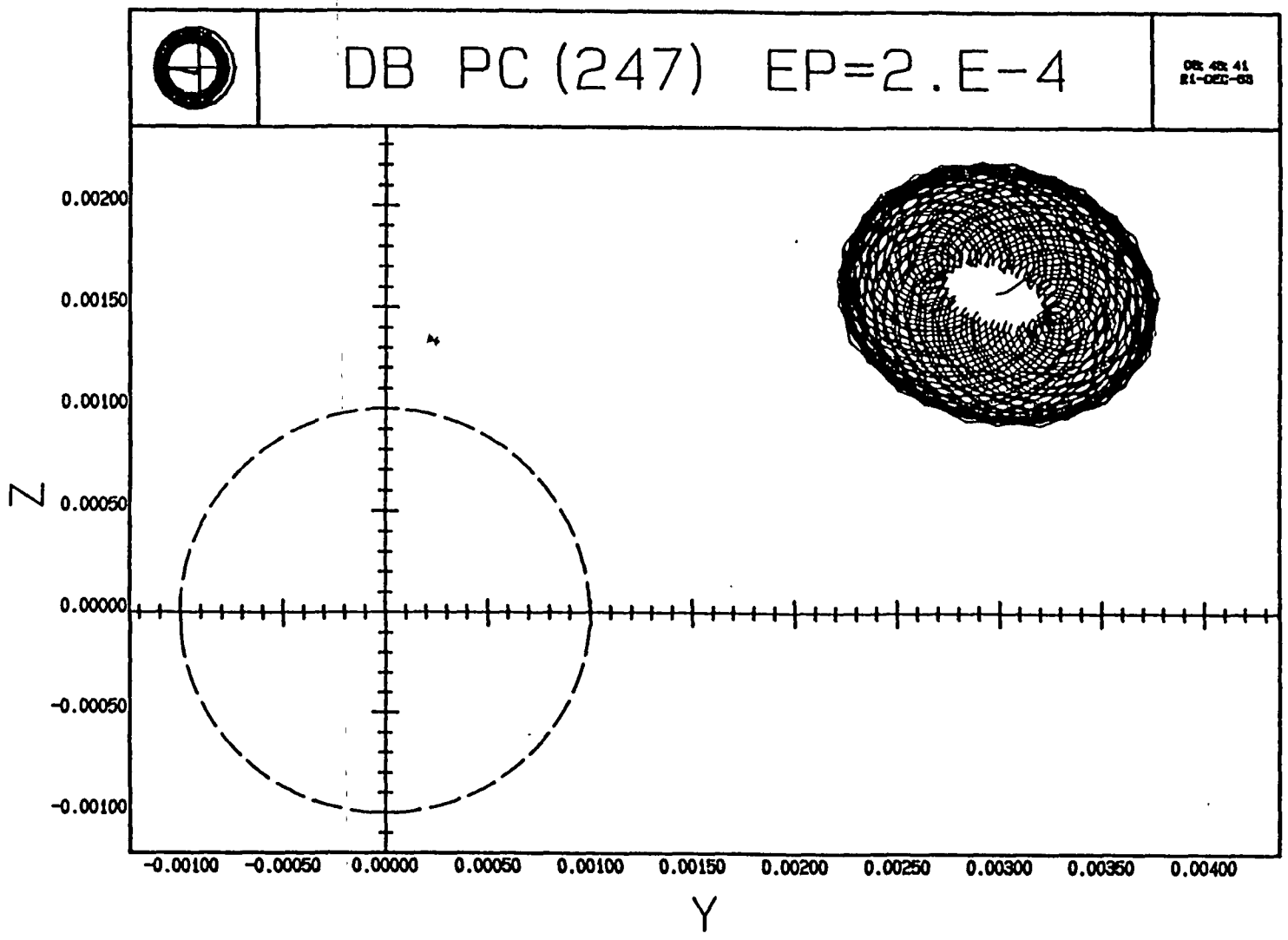

Figure 3.21a Stable Orbit Around Equilibrium Point for Test Case 2.

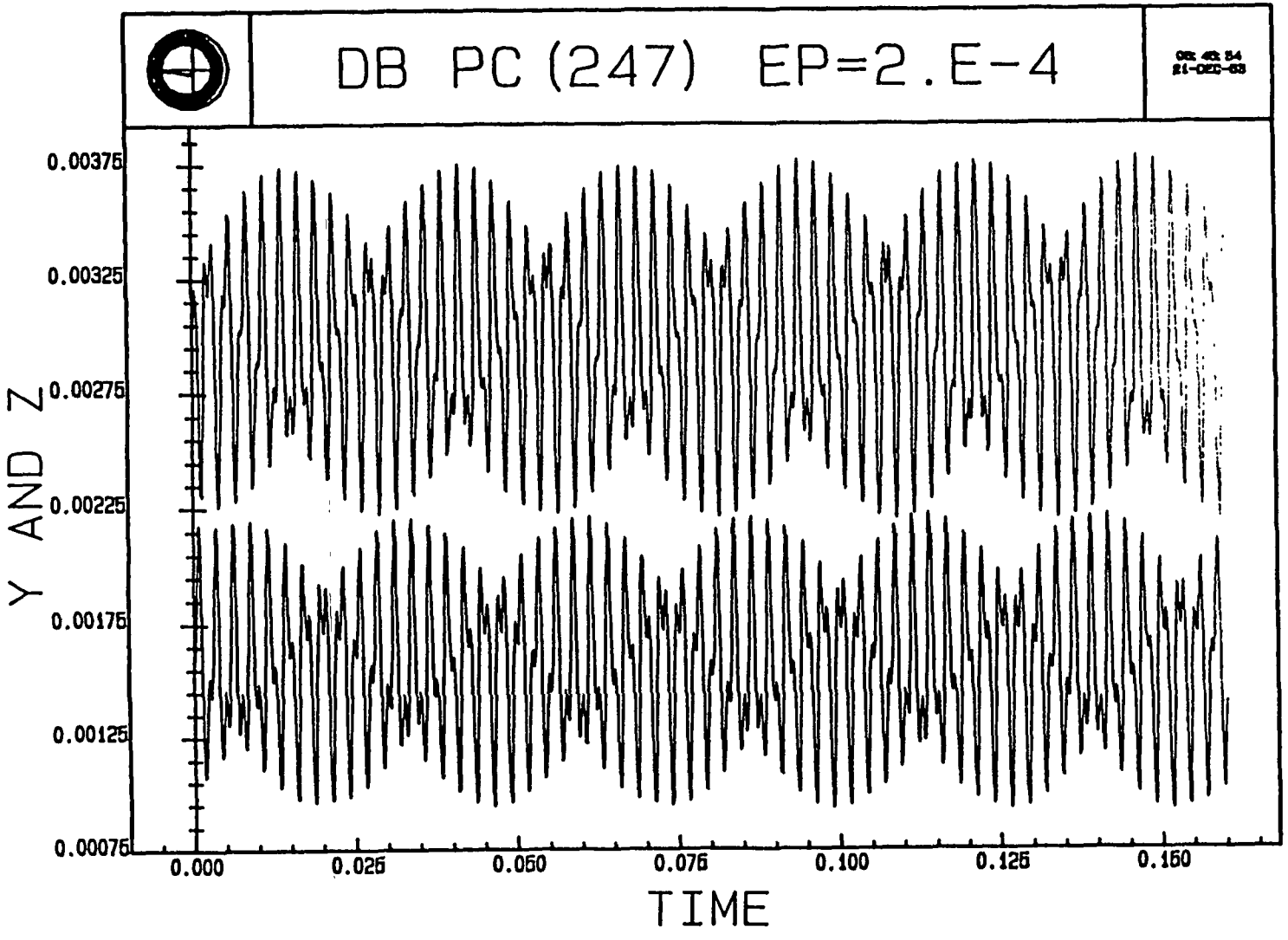

Figure 3.21b The x and y Components as a Function of Time.

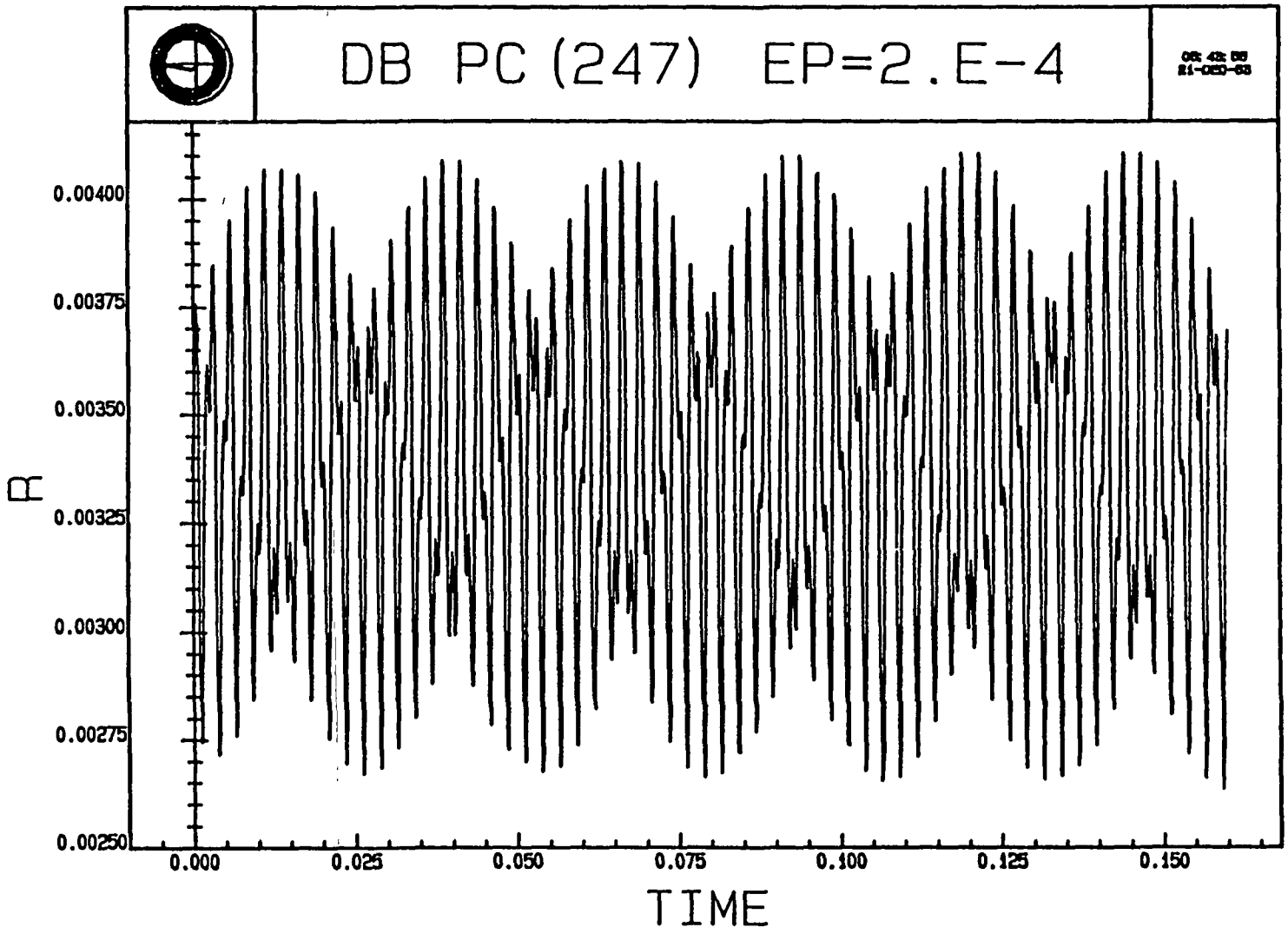

Figure 3.21c The Orbit Radius as a Function of Time.

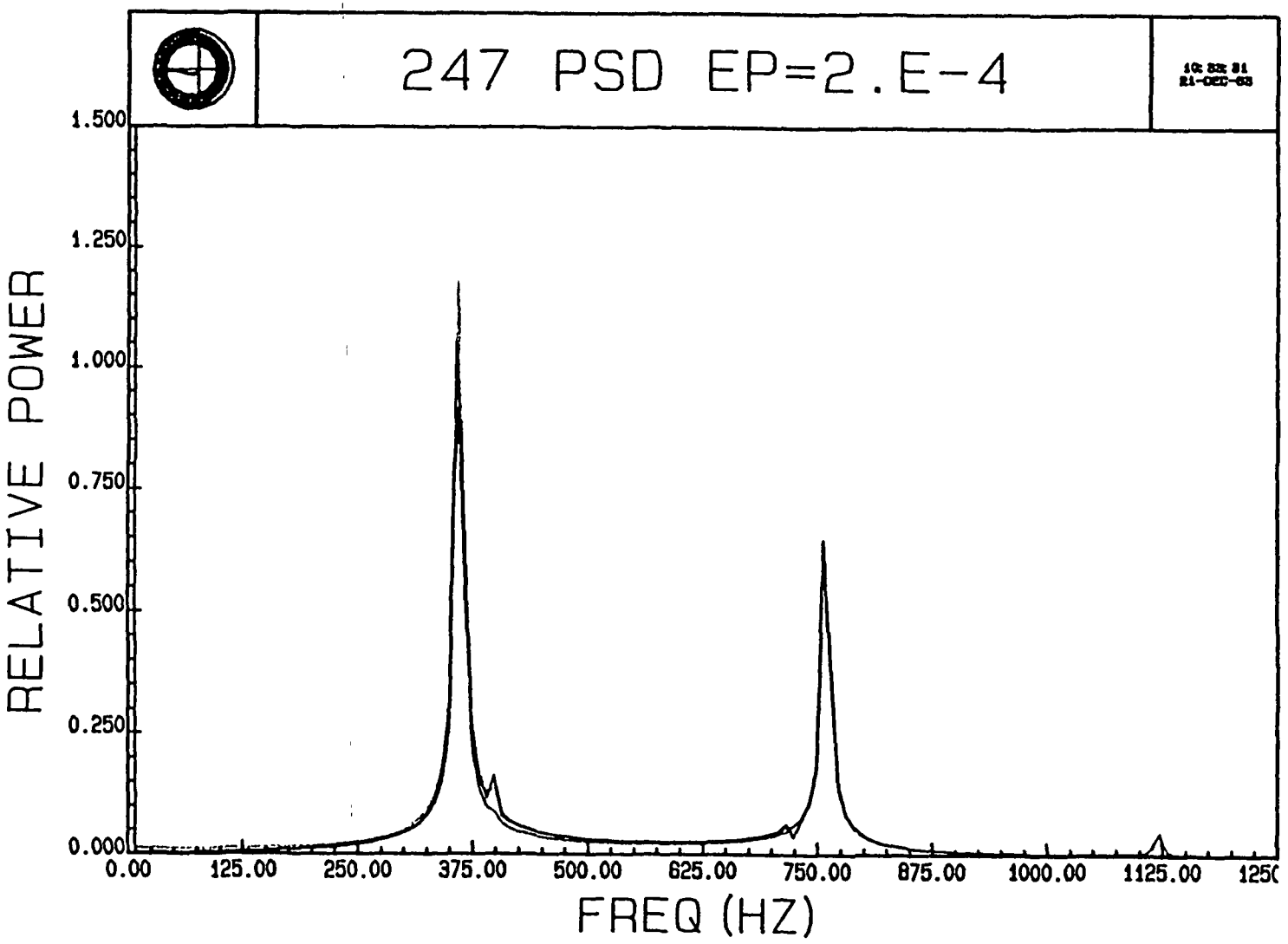

Figure 3.21d PSD of the Unstable Orbit; Shaft Speed: 760 Hz.

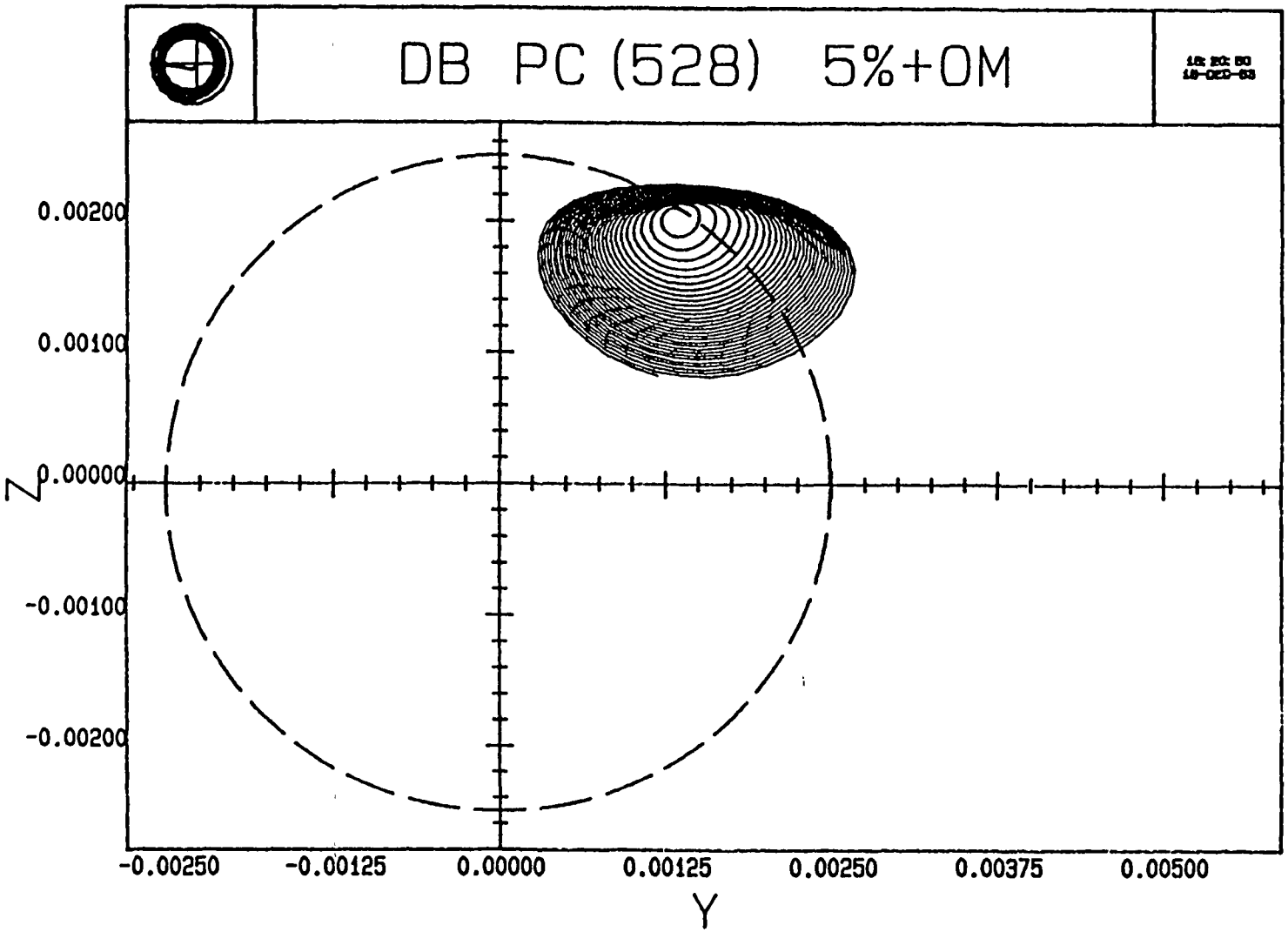

Figure 3.22 Unstable Orbit Around Equilibrium Point for Test Case 4.

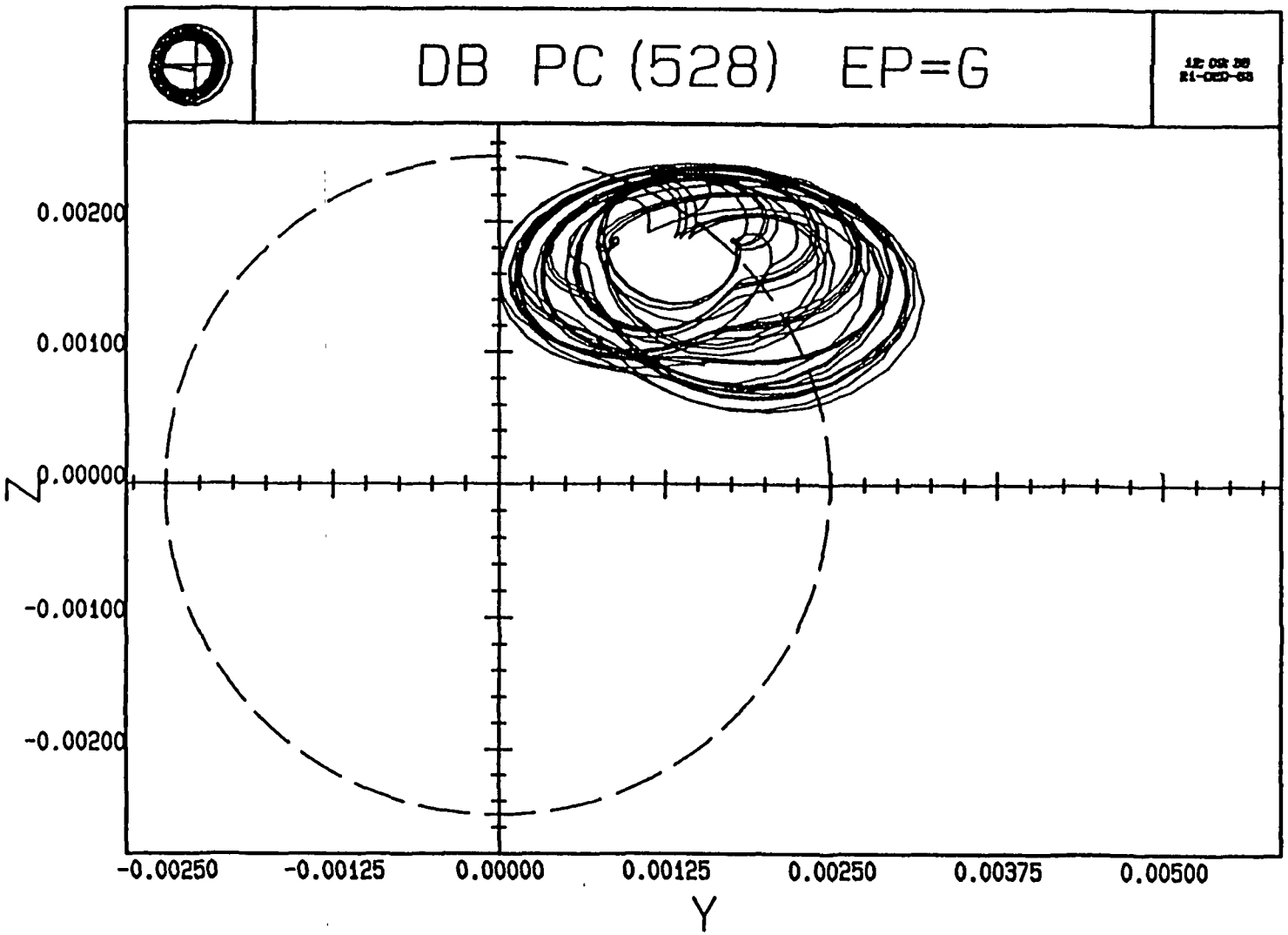

Figure 3.23a Stable Orbit Around Equilibrium Point for Test Case 4.

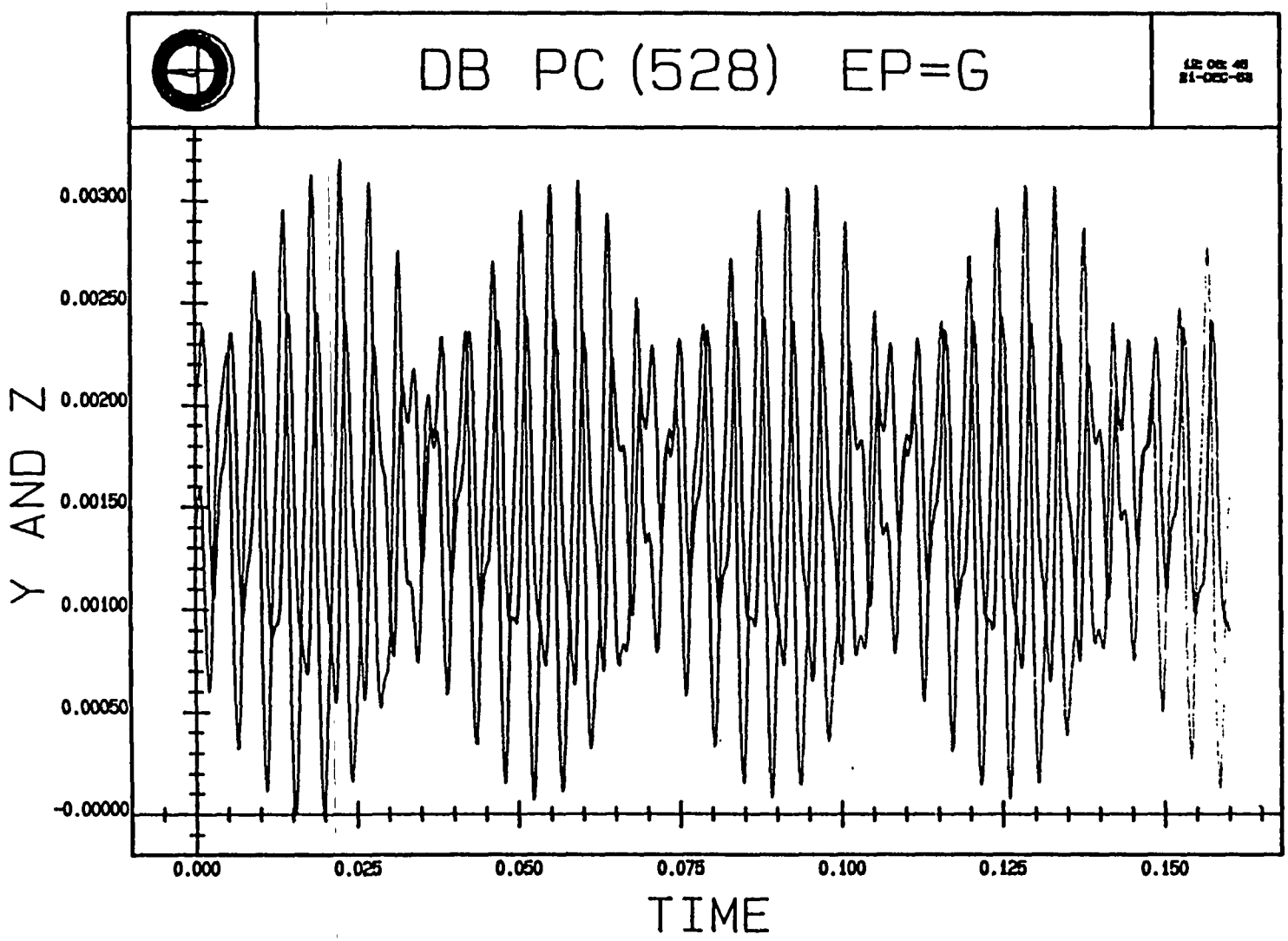

Figure 3.23b The x and y Components as a Function of Time.

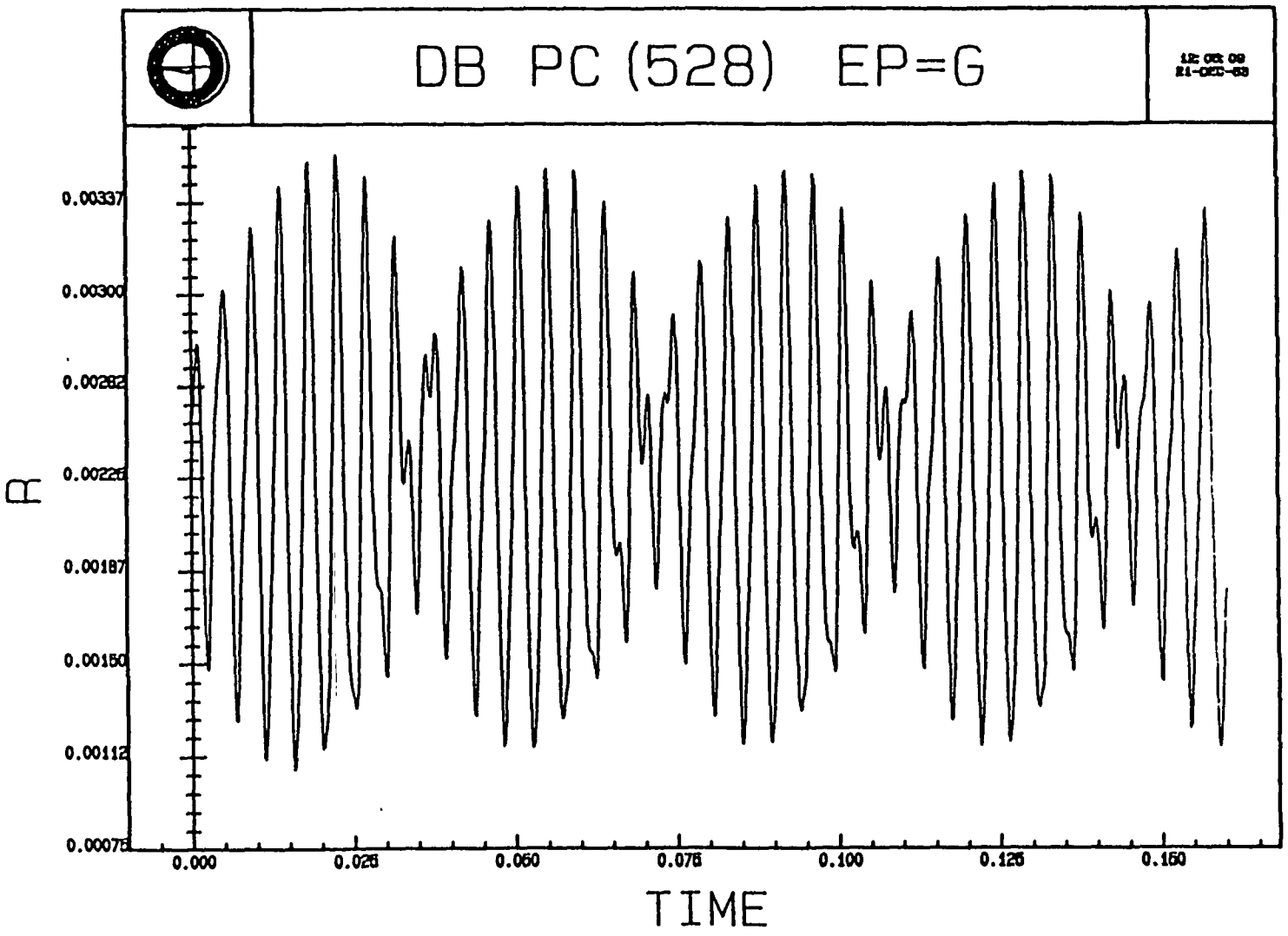

Figure 3.23c The Orbit Radius as a Function of Time.

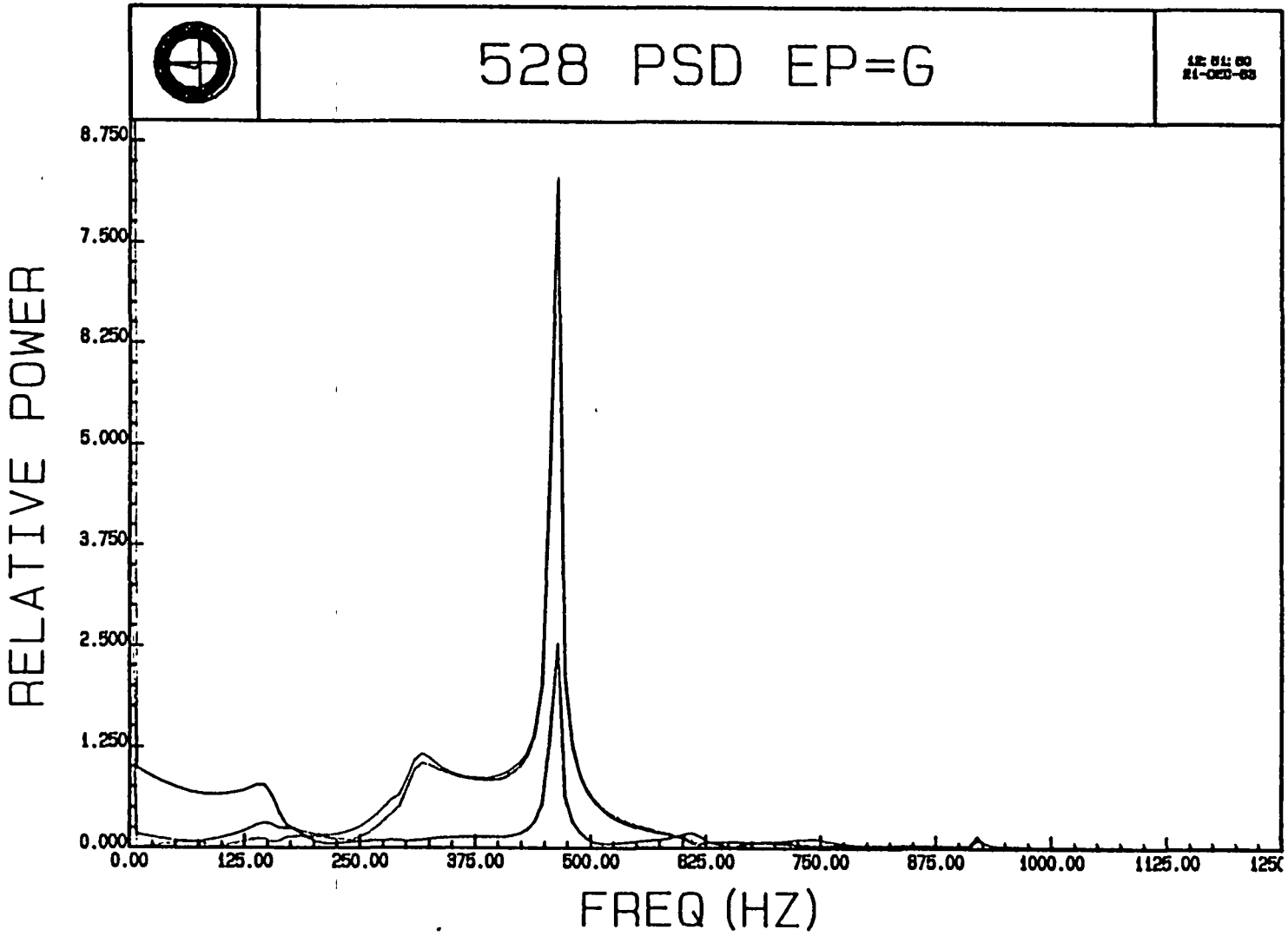

Figure 3.23d PSD of the Stable Orbit; Shaft Speed: 760 Hz.

4.0 BEARING LOADS CONSIDERATIONS

One of the major points of the study is to determine the effects of the system parameters on bearing loads. If these loads become too large, the effects are detrimental. Initially, we will look at this matter using a system with no side forces present, but with a rotor imbalance. We will then take into consideration the added effects of a side force present in the system.

Recall that for the parameters describing our rotor model, no equilibrium orbits exist in the frequency range of interest, that is, in the vicinity of 500 Hertz, when no side force is present. One may wish to refer back to section 3.3. Therefore, combinations of deadbands and offsets which produce equilibrium orbits are used to examine the effects of rotor eccentricity and deadband on bearing loads.

Plotted in Figure 4.1 are the bearing loads which result when the rotor eccentricity is 0.1 mils. The deadband range is from 0.0 to 0.2 mils. The general behavior is that the smaller deadband produces the largest bearing load. This makes sense because the seal forces must be overcome before there is any interaction between the rotor and the bearings. The more distance between the rotor and the bearing there is, the more effect the seal forces have.

Figure 4.2 is a plot of the bearing loads with a rotor imbalance of 0.2 mils. The bearing forces are twice the magnitude of those just examined. the range of deadbands are from zero to 0.4 mils. The non-existance phenomenon for the equilibrium orbit is clearly seen in these two plots by the way some of the curves suddenly drop off to zero, or suddenly appear.

When the deadbands and imbalances defined in chapter 2 are used, with zero side force, the bearing loads are as plotted in Figure 4.3. As the frequency increases, the rotor displacements grow and the bearing loads become quite large. Figure 4.3 is an enlargement of 4.3 to facilitate the examination of the zero deadband curve. The peak of the bearing load curve for this case occurs at the system natural frequency, as would be expected.

Bearing load analysis is performed for two side force values. A value of side force is chosen so that it is always greater than F_{SMIN} for any of the five deadbands considered up to a frequency of 5000 radians per second. Figure 4.4 is a plot of the bearing loads for the deadbands of 0.5 to 2.5 mils for the side force of 1350 pounds. The maximum bearing loads occur at the system natural frequency of approximately 2424 radians/second, the smallest deadband producing the largest load. The load curves are plotted only up to a shaft spin frequency of 4000 radians because the system becomes unstable for frequencies higher than that. The presence of the rotor eccentricity of 0.2 mils is responsible for the unstable behavior.

A similar family of curves is produced when the side force is increased to twice that of the "minimum" side force, or, 2700 pounds. The plots of these bearing loads are given in Figure 4.5. The same general behavior is exhibited as before. Instability occurs somewhat sooner, at 3400 radians/second. The loads are much greater, as well.

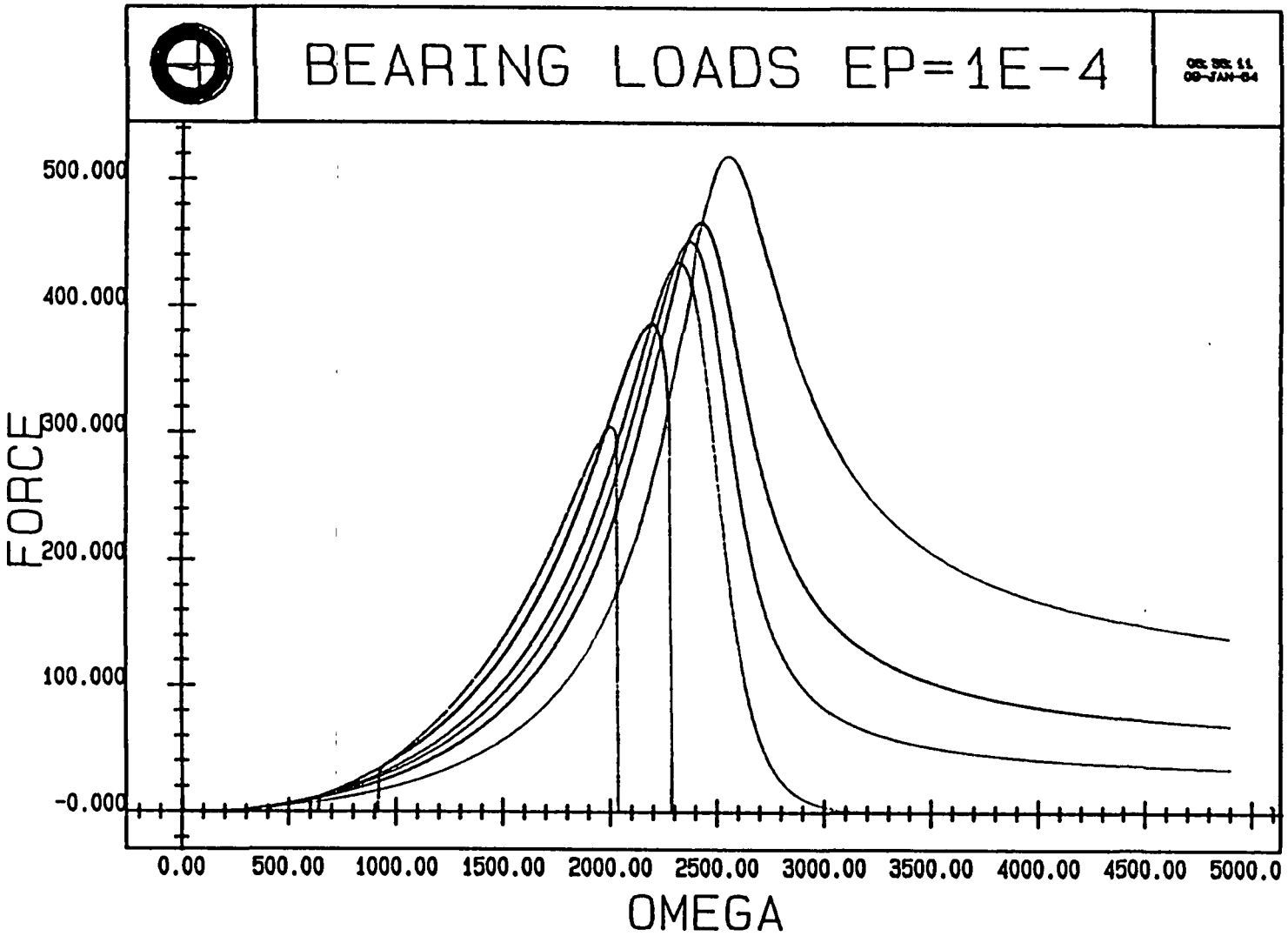

Figure 4.1 Bearing Loads for a Rotor Eccentricity of 0.1 Mils.

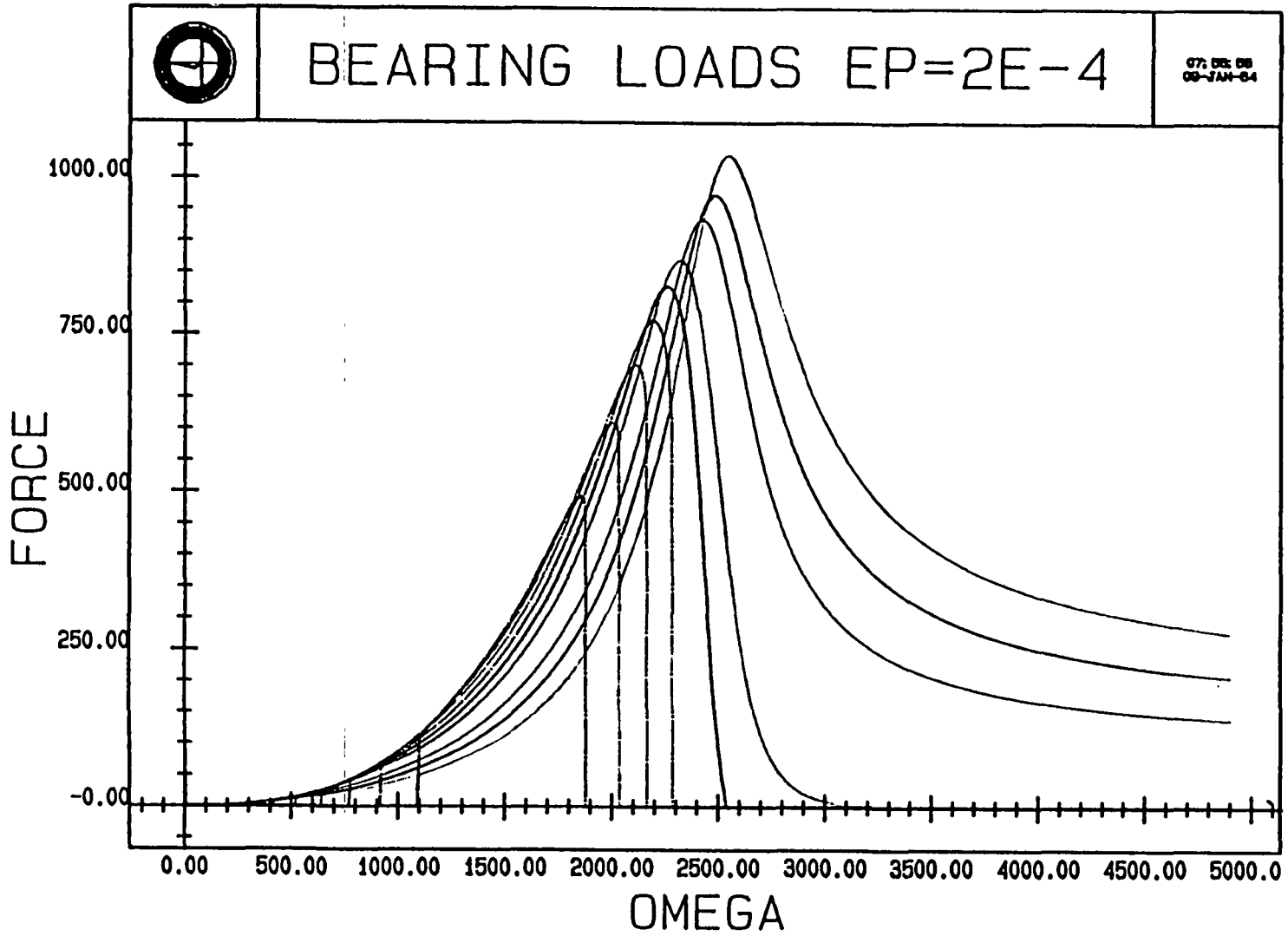

Figure 4.2 Bearing Loads for a Rotor Eccentricity of 0.2 Mils.

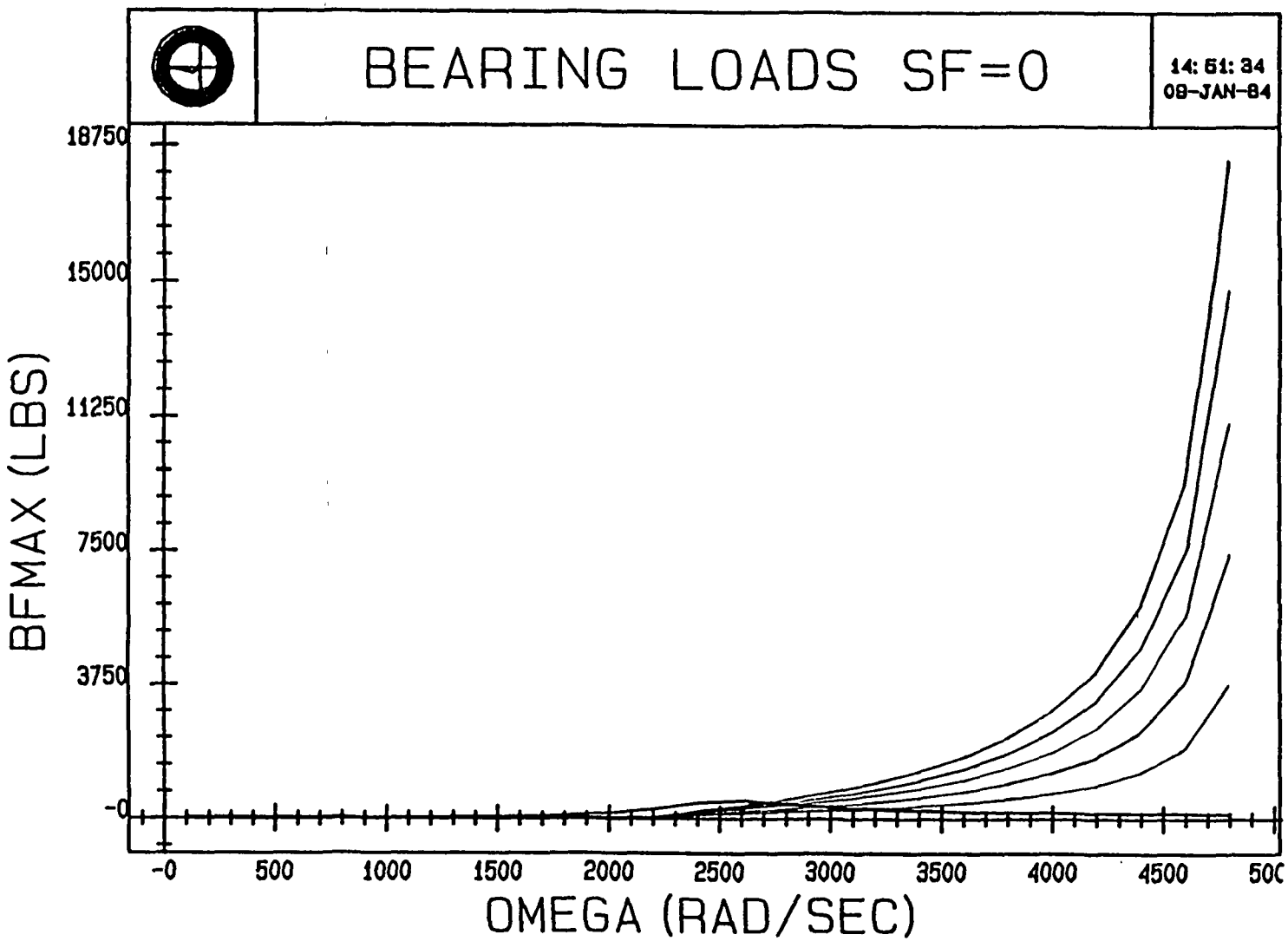

Figure 4.3a Bearing Loads for a Zero Side Force.

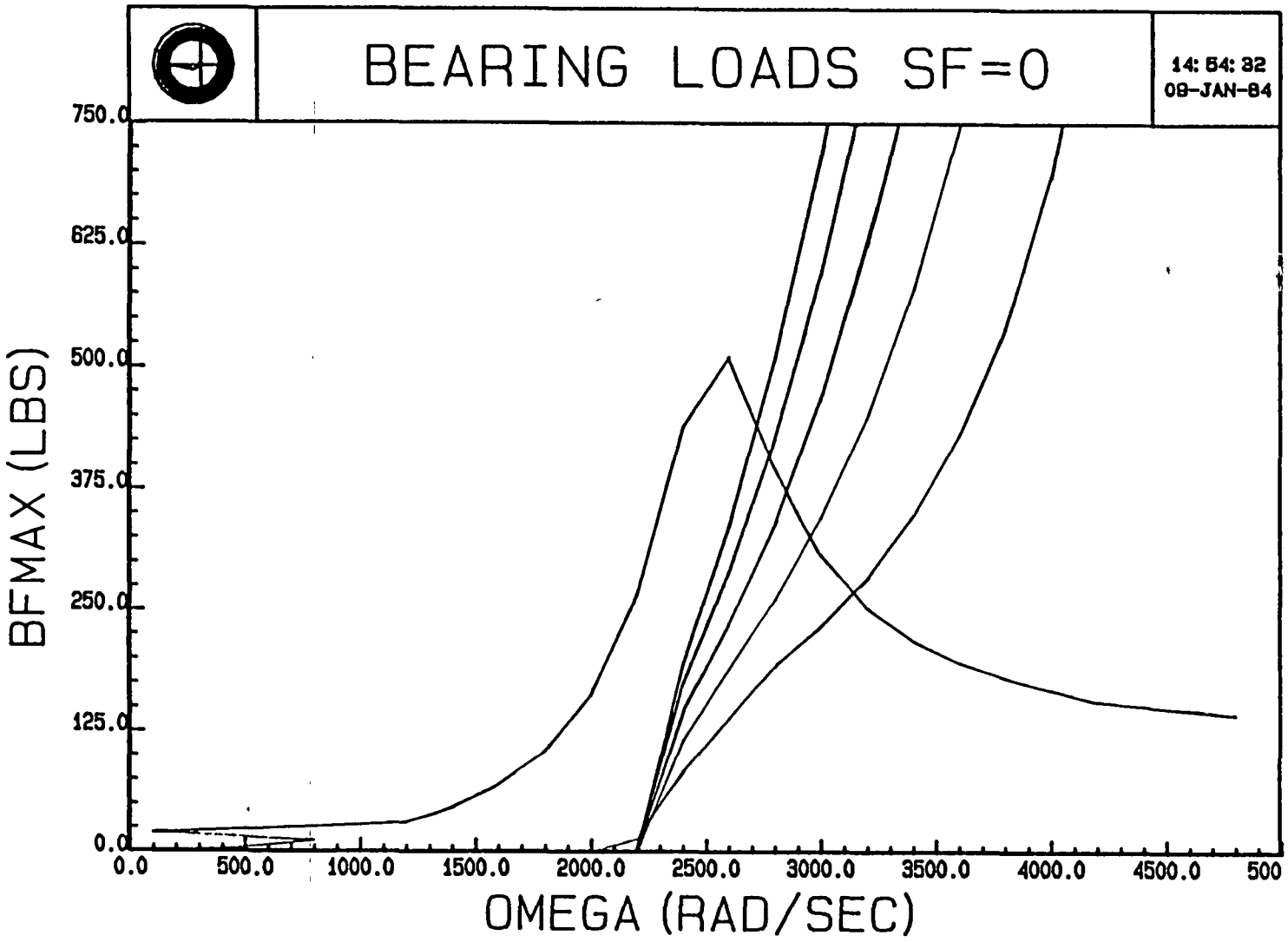

Figure 4.3b Enlargement of 4.3a to Indicate the Zero Deadband Curve.

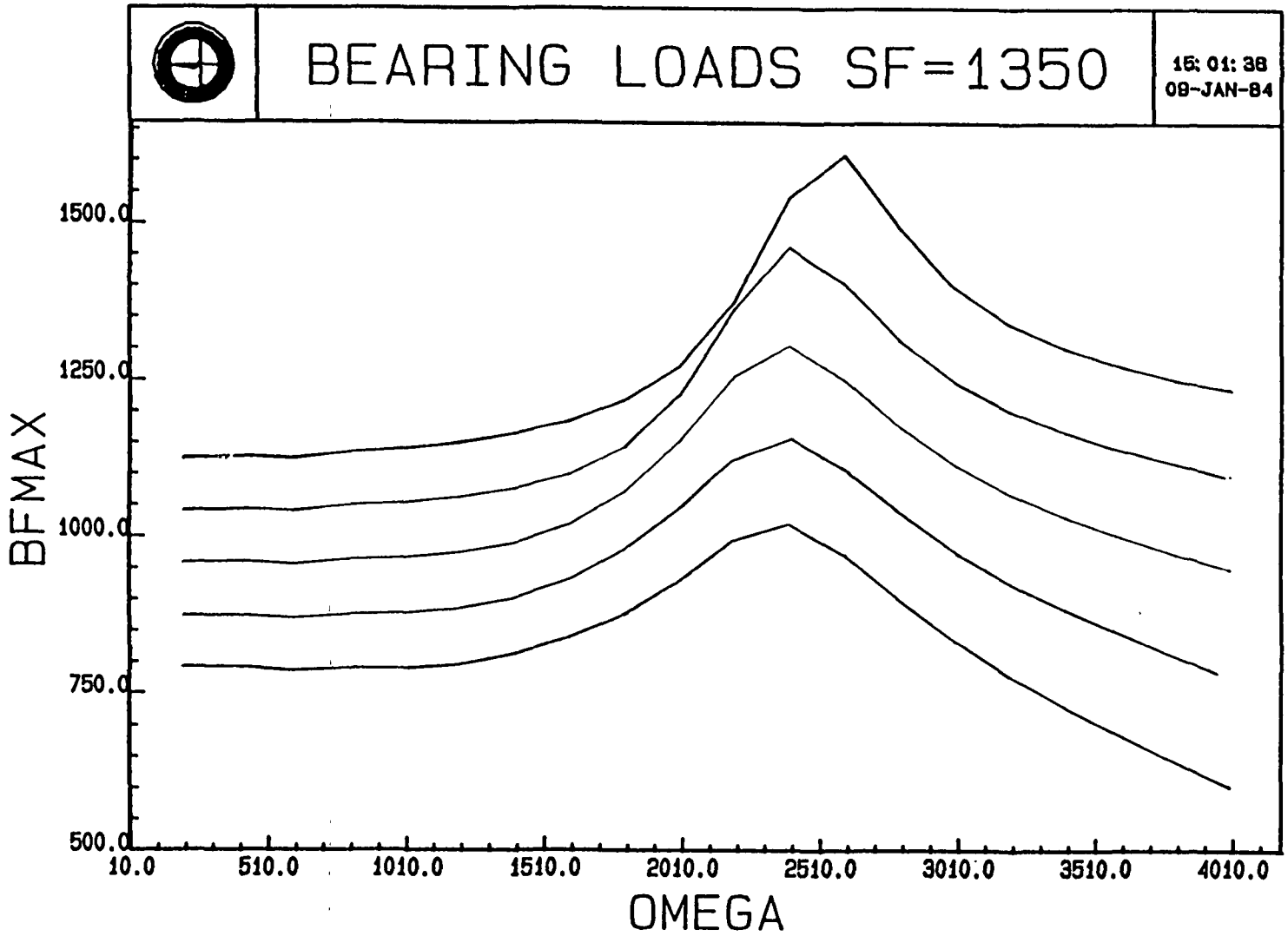

Figure 4.4 Bearing Loads for a Side Force of 1350 Pounds.

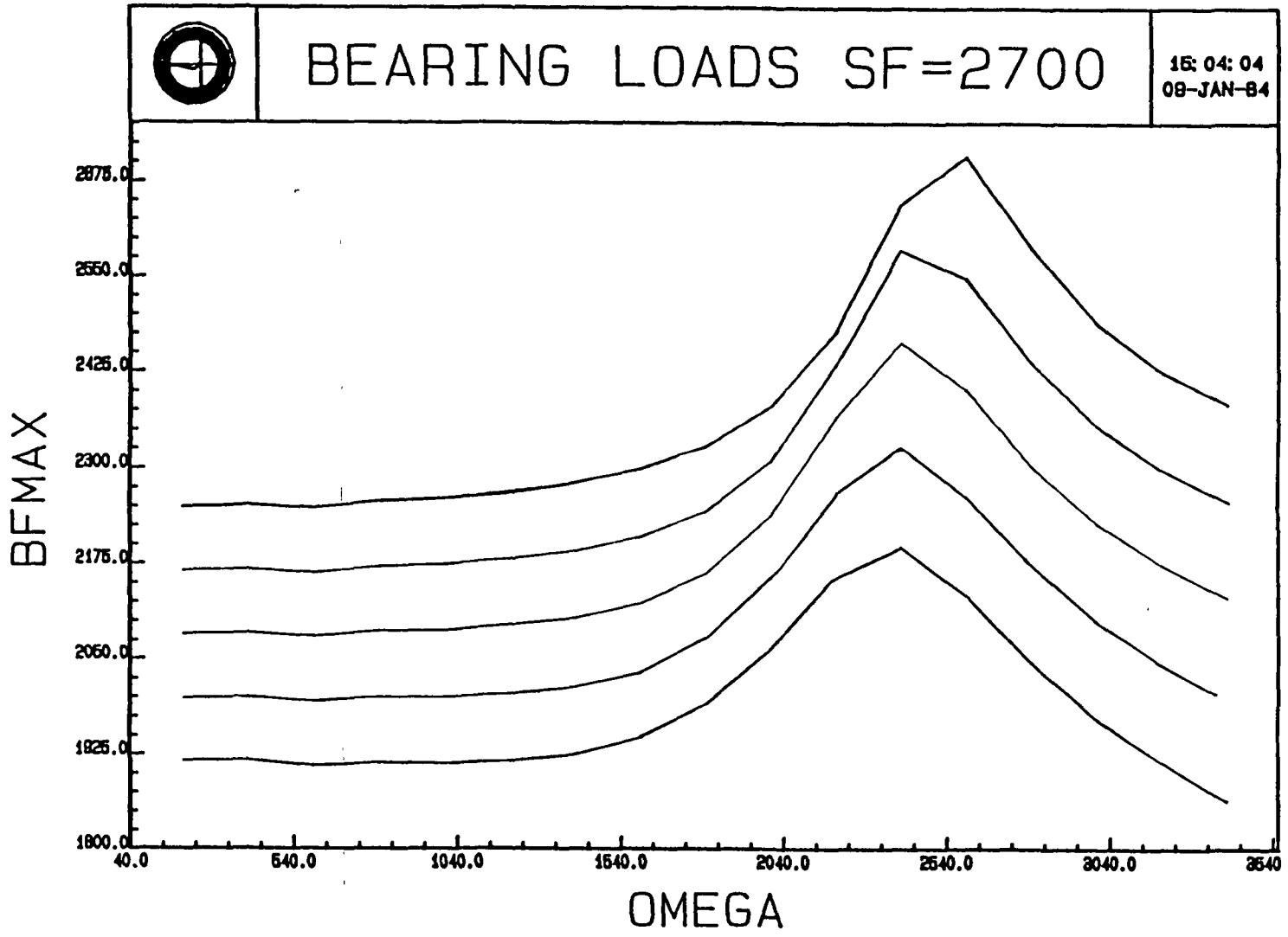

Figure 4.5 Bearing Loads for a Side Force of 2700 Pounds.

The curves presented were generated using the simulation to determine the maximum rotor displacement, after steady-state is achieved. Having this value, it is a simple matter to compute the bearing load. Equation 4.1 is used.

$$BF_{MAX} = K_B(r_{MAX} - g) \tag{4.1}$$

5.0 SOFTWARE DEVELOPED FOR THE STUDY

In the course of investigating the effects of the deadband on stability, a number of computer programs have been developed to aid in the analysis. This section will outline the software developed with the program listings included at the end of the section.

Two versions of the simple rotor model simulation have been written, one in polar coordinates and one in Cartesian coordinates. Each main program is an integration loop driver which calls a subroutine containing the equations of motion describing the system. The program POLRNG.FOR is the polar coordinates main driver program, with the associated subroutine contained in POLAR.FOR. PRECDB.FOR is the Cartesian coordinate driver program with the equations of motion contained in the subroutine whose program name is PREC.FOR. The power spectral densities are computed from data files which is constructed by the above mentioned routines using the programs PSD.FOR and PFFT.FOR. The Cooley-Tukey method for performing the Fast Fourier Transform is used.

The convergence algorithm is the next listing. The name of the main program is PIPC.FOR. Two of the five associated subroutines, a Runge-Kutta integration routine and the associated equations of motion are contained in the program PRK4.FOR. The remaining three, a matrix inversion routine, a matrix times a vector routine, and a matrix write routine are included in the program PMTRX.FOR. PIPC.FOR is the algorithm used for determining the initial conditions for C-type motion. It is identical to the program used for A-type motion, except that the value 2π is not subtracted from the third state vector, Y(3) in the program. The lines in which this computation is performed are preceeded with a right arrow (--->).

Several programs were composed to aid in the analysis of the stability of the system. The major functions of these programs are to compute equilibrium points and radii, to root resulting polynomials, and to generate the coefficients of various characteristic equations. The program PQUAD.FOR is used in the analysis of the system in which the deadband, side forces, and rotor eccentricity are all zero. This program computes the roots of the characteristic equation. The programs PQ4.FOR and PQC.FOR are used for analysis of the system which has only a deadband. PQ4 computes the equilibrium radius along with the coefficients of the system characteristic equation. Once the coefficients are determined, standard routine algorithms are used to obtain the stability boundaries. PQC computes the coefficients of the third order system characteristic equation. The routines PREP.FOR and PREPBF.FOR were generated to compute the equilibrium radius as a function of frequency and the resulting bearing force, respectively, for the system in which the deadband and the rotor eccentricity are not zero. The program PSRO.FOR is used to determine the equilibrium point for a system in which the deadband and sideforces are non-zero. PSROOT.FOR is used to determine the stability boundaries by rooting the characteristic equation which has been cast into the form of two quadratic equations for the non-zero side force system. PNROOT.FOR is used to establish the stability boundary for the non-dimensionalized system.

For each program listed, a header is included to define the program function.

```
C  FILE NAME:  POLRNG.FOR       28-APR-83              APW
C
C               POLAR COORDINATES VERSION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       RUNGE-KUTTA INTEGRATION LOOP DRIVER.
C       THIS PROGRAM ASSUMES THE DERIVATIVE FUNCTION YPR(T,Y) IS
C       DEFINED EXTERNALLY BY A SUBROUTINE CALLED YPR.  THE DATA
C       IS ASSUMED TO BE PASSED THROUGH THE CALL STATEMENT IN THE
C       FORM CALL YPR(N,T,Y,YD); WHERE N IS THE DIMENSION OF THE
C       STATE VECTOR, T IS THE INDEPENDENT VARIABLE (TIME), Y IS
C       THE STATE VECTOR AND YD IS THE DERIVATIVE OF THE STATE
C       VECTOR.  INPUTS TO THE PROGRAM ARE FROM THE TERMINAL OR
C       A COMMAND FILE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       DIMENSION Y(40),YD(40),YD0(40),YD1(40),YD2(40),YD3(40),TEMP(40)
       COMMON /NONLIN/ YNL2,YNL4,FB,FS,CB,SB,CS2,CS4,QS
       COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE,RSIDE,PHSIDE,CQ
CCC
C  INITIALIZE SIMULATION FROM COMMAND FILE INPUT
CCC
       TYPE *,'INPUT THE INTEGRATION START AND STOP TIMES.'
       ACCEPT *,T,TSTOP
       TYPE *,'INPUT THE TIME STEP AND NUMBER OF STEPS PER OUTPUT.'
       ACCEPT *,H,NPR
       IPR=NPR
       TYPE *,'INPUT THE DIMENSION OF THE STATE VECTOR Y.'
       ACCEPT *,N
       TYPE *,'INPUT THE INITIAL STATE VECTOR.'
       ACCEPT *,(Y(I),I=1,N)
CCC
C      SSME ROTOR MODEL INITIALIZATION OF PARAMETERS
CCC
      TYPE *,'Enter the FFT data recording start time:'
      ACCEPT *,TFFT
      TYPE *,'Enter values for the following quantities on separate'
      TYPE *,'lines: RM, BK, SK, CS, G, EP, OMEGA, FSIDE,& CQ'
      ACCEPT *,RM
      ACCEPT *,BK
      ACCEPT *,SK
      ACCEPT *,CS
      ACCEPT *,G
      ACCEPT *,EP
      ACCEPT *,OMEGA
      ACCEPT *,FSIDE
      ACCEPT *,CQ
        QS = (CS*OMEGA)/(2.0*RM)
        FSIDE = FSIDE/RM
        HO6=H/6.
        HO2=H/2.
        ICOUNT = 0.
        DBY = G
        DBZ = 0.
```

```
        DBR = G
        TREC = TSTOP - .005
        IF(H.EQ.0)GO TO 9999
        DIV = AINT((TSTOP - T)/(H*FLOAT(NPR))) + 2.
        GO TO 9991
9999    DIV=1.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
9991    OPEN (UNIT=1,TYPE='NEW')
        OPEN (UNIT=2,TYPE='NEW')
        OPEN (UNIT=3,TYPE='NEW')
        OPEN (UNIT=4,TYPE='NEW')
        CALL HORIZ(16,4)
        Q1 = Y(1)*COS(Y(3))
        Q2 = Y(1)*SIN(Y(3))
CCC
C       WRITE INITIAL TIME AND STATES TO PLOT FILES.
CCC
        WRITE (2,11) T,Y(1),Y(3),Q1,Q2,DBY,DBZ
        ICOUNT = ICOUNT + 1
CCC
C       TOP OF INTEGRATION LOOP.
CCC
1       CONTINUE
        CALL YPR(N,T,Y,YD0)
        T=T+H02
        DO 1000 I=1,N
1000    TEMP(I)=Y(I)+H02*YD0(I)
        CALL YPR(N,T,TEMP,YD1)
        DO 1010 I=1,N
1010    TEMP(I)=Y(I)+H02*YD1(I)
        CALL YPR(N,T,TEMP,YD2)
        T=T+H02
        DO 1020 I=1,N
1020    TEMP(I)=Y(I)+H*YD2(I)
        CALL YPR(N,T,TEMP,YD3)
        DO 1030 I=1,N
1030    Y(I)=Y(I)+H06*(YD0(I)+YD3(I)+2*(YD1(I)+YD2(I)))
        Q1 = Y(1)*COS(Y(3))
        Q2 = Y(1)*SIN(Y(3))
CCC
C  WRITE Y(1) TO DATA FILE FOR USE IN FFT ROUTINE.
CCC
        IF (T.LT.TFFT) GOTO 1041
        WRITE(1,10) Y(1),Q1,Q2
        KOUNT=KOUNT+1
CC (TREC ADDED 14-NOV-83)
C  WRITE THE FOUR STATES TO A DATA FILE ONCE THE THING HAS
C  SETTLED DOWN.
CC
1041    IF (T.LT.TREC) GO TO 1040
        WRITE(4,*)T,Y(1),Y(2),Y(3),Y(4),YD3(2),YD3(4)
CCC
C       TEST WHETHER RESULTS ARE TO BE OUTPUT.
CCC
1040    IPR=IPR-1
```

```
        IF (IPR.GT.0) GOTO 1
        IPR=NPR
CCC
C  THE SUBROUTINE DEAD COMPUTES FOR PLOTTING A CIRCLE OF RADIUS EQUAL TO
C  THE DEADBAND.
CCC
        CALL DEAD(DBY,DBZ,DBR,ICOUNT,DIV)
        WRITE (2,11) T,Y(1),Y(3),Q1,Q2,DBY,DBZ
        ICOUNT = ICOUNT + 1
        IF (T.LT.TSTOP) GOTO 1
        ICOL = 10
        WRITE(3,12) ICOL,ICOUNT
        WRITE(3,13) 'TIME'
        WRITE(3,13) 'R'
        WRITE(3,13) 'PHI'
        WRITE(3,13) 'Y'
        WRITE(3,13) 'Z'
        WRITE(3,13) 'DBY'
        WRITE(3,13) 'DBZ'
        WRITE(3,13) 'FSIDE'
        WRITE(3,13) 'RSIDE'
        WRITE(3,13) 'PHSIDE'
11      FORMAT(7E11.4)
12      FORMAT(2I5)
13      FORMAT(A8)
        CLOSE (UNIT=1)
        CLOSE (UNIT=2)
        CLOSE (UNIT=3)
        CALL HORIZ(10,4)
        CLOSE (UNIT=4)
        TYPE*,'#PTS. IN DATA FILE:',KOUNT
        END

        SUBROUTINE DEAD(Y,Z,R,M,DIV)
        ANGLE = (6.283185308/DIV)*FLOAT(M)
        Y = R*COS(ANGLE)
        Z = R*SIN(ANGLE)
        RETURN
        END


      SUBROUTINE HORIZ(I,N)
      IF ( I .EQ. 10 ) GOTO 100
      IF ( I .EQ. 16 ) GOTO 130
10    FORMAT(10A1)
100   WRITE(N,10) 27,91,49,119
      GOTO 200
130   WRITE(N,10) 27,91,52,119
      GOTO 200
200   CONTINUE
      RETURN
      END
```

```
C-------------------------------------------------------------------------
C    file name:  POLAR.FOR              26-APR-83                 APW
C
C  The equations given below describe the SIMPLE SSME deadband rotor model...
C
C               POLAR COORDINATES VERSION
C
      SUBROUTINE YPR(N,T,Y,YD)
      DIMENSION Y(4),YD(4)
      COMMON /NONLIN/ YNL2,YNL4,FB,FS,CB,SB,CS2,CS4,QS
      COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE,RSIDE,PHSIDE,CQ
C
C  Enter the equations here
C
      IF (Y(1).GT.G) GO TO 100
        FB = 0.
      GO TO 200
100   FB = - BK*(Y(1) - G)/RM
200   FS = - SK*Y(1)/RM
      YNL2 = Y(1)*(Y(4)**2)
      YNL4 = - (2.0*Y(2)*Y(4))/Y(1)
      CB = EP*(OMEGA**2)*COS(OMEGA*T - Y(3))
      SB = ((OMEGA**2)*EP*SIN(OMEGA*T - Y(3)))/Y(1)
      RSIDE = FSIDE*COS(Y(3))
      PHSIDE = -(FSIDE*SIN(Y(3)))/Y(1)
      CS2 = - (CS*Y(2))/RM
      CS4 = - (CS*Y(4))/RM
      CQ2 = - (CQ*Y(1)*Y(4))/RM
      CQ4 =   (CQ*Y(2))/(RM*Y(1))
      YD(1) = Y(2)
      YD(2) =  FB + FS + CS2 + CQ2 + YNL2 + CB + RSIDE
      YD(3) = Y(4)
      YD(4) = CS4 + CQ4 + QS + YNL4 + SB + PHSIDE
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C  FILE NAME:  PRECDB.FOR        28-APR-83              APW
C
C       RECTANGULAR COORDINATES VERSION
C
C-----------------------------------------------------------------------
C       RUNGE-KUTTA INTEGRATION LOOP DRIVER.
C       THIS PROGRAM ASSUMES THE DERIVATIVE FUNCTION YPR(T,Y) IS
C       DEFINED EXTERNALLY BY A SUBROUTINE CALLED YPR.  THE DATA
C       IS ASSUMED TO BE PASSED THROUGH THE CALL STATEMENT IN THE
C       FORM CALL YPR(N,T,Y,YD); WHERE N IS THE DIMENSION OF THE
C       STATE VECTOR, T IS THE INDEPENDENT VARIABLE (TIME), Y IS
C       THE STATE VECTOR AND YD IS THE DERIVATIVE OF THE STATE
C       VECTOR.  INPUTS TO THE PROGRAM ARE FROM THE TERMINAL OR
C       A COMMAND FILE.
C-----------------------------------------------------------------------

      DIMENSION Y(40),YD(40),YD0(40),YD1(40),YD2(40),YD3(40),TEMP(40)
      COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE
      TYPE *,'INPUT THE INTEGRATION START AND STOP TIMES.'
      ACCEPT *,T,TSTOP
      TYPE *,'INPUT THE TIME STEP AND NUMBER OF STEPS PER OUTPUT.'
      ACCEPT *,H,NPR
      TYPE *,'INPUT THE DIMENSION OF THE STATE VECTOR Y.'
      ACCEPT *,N
      TYPE *,'INPUT THE INITIAL STATE VECTOR.'
      ACCEPT *,(Y(I),I=1,N)
C
C       This section added for SSME model for initialization of parameters
C
      TYPE *,'Enter plot start time:'
      ACCEPT *,TPLOT
      TYPE *,'Enter values for the following quantities on separate'
      TYPE *,'lines: RM, BK, SK, CS, G, EP, OMEGA, & FSIDE'
      ACCEPT *,RM
      ACCEPT *,BK
      ACCEPT *,SK
      ACCEPT *,CS
      ACCEPT *,G
      ACCEPT *,EP
      ACCEPT *,OMEGA
      ACCEPT *,FSIDE
        HO6=H/6.
        HO2=H/2.
        YMAX=0.
        YMIN=0.
        ICOUNT = 0.
        DBY = G
        DBZ = 0.
        DBR = G
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        OPEN (UNIT=1,TYPE='NEW')
        OPEN (UNIT=2,TYPE='NEW')
        OPEN (UNIT=3,TYPE='NEW')
```

```
C
C       WRITE INITIAL TIME AND STATE.
C
        IF(TPLOT.GT.0.0) GO TO 1
        WRITE (2,11) T,Y(1),Y(3),DBY,DBZ
        ICOUNT = ICOUNT + 1
10      FORMAT(3(E15.8))
11      FORMAT(6E11.4)
        IPR=NPR
C
C       TOP OF INTEGRATION LOOP.
C
1       CONTINUE
        CALL YPR(N,T,Y,YDO)
        T=T+HO2
        DO 1000 I=1,N
1000    TEMP(I)=Y(I)+HO2*YDO(I)
        CALL YPR(N,T,TEMP,YD1)
        DO 1010 I=1,N
1010    TEMP(I)=Y(I)+HO2*YD1(I)
        CALL YPR(N,T,TEMP,YD2)
        T=T+HO2
        DO 1020 I=1,N
1020    TEMP(I)=Y(I)+H*YD2(I)
        CALL YPR(N,T,TEMP,YD3)
        DO 1030 I=1,N
1030    Y(I)=Y(I)+HO6*(YDO(I)+YD3(I)+2*(YD1(I)+YD2(I)))
        RAD = SQRT(Y(1)**2 + Y(3)**2)
        IF (T.LT.TPLOT) GO TO 1040
        WRITE(1,10)RAD,Y(1),Y(3)
        KOUNT = KOUNT + 1
C
C       TEST WHETHER RESULTS ARE TO BE OUTPUT.
C
1040    IPR=IPR-1
        IF (IPR.GT.0) GOTO 1
        IPR=NPR
C
C  THE SUBROUTINE DEAD COMPUTES FOR PLOTTING A CIRCLE OF RADIUS EQUAL TO
C  THE DEADBAND.
C
        IF (T.LT.TPLOT) GO TO 1
        CALL DEAD(DBY,DBZ,DBR,ICOUNT)
        WRITE (2,11) T,Y(1),Y(3),DBY,DBZ,RAD
        ICOUNT = ICOUNT + 1
        IF (T.LT.TSTOP) GOTO 1
        ICOL = 6
        WRITE(3,12) ICOL,ICOUNT
        WRITE(3,13) 'TIME'
        WRITE(3,13) 'Q1'
        WRITE(3,13) 'Q3'
        WRITE(3,13) 'DBY'
```

```
      WRITE(3,13) 'DBZ'
      WRITE(3,13) 'RAD'
12    FORMAT(2I5)
13    FORMAT(A8)
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)
      TYPE*,'# OF POINTS IN DATA FILE:',KOUNT
      END


      SUBROUTINE DEAD(Y,Z,R,M)
      ANGLE = (6.283185308/502.0)*FLOAT(M)
      Y = R*COS(ANGLE)
      Z = R*SIN(ANGLE)
      RETURN
      END
```

```
C  file name:  PREC.FOR          26-APR-83                APW
C
C  The equations given below describe the SIMPLE SSME deadband rotor model...
C
C                CARTESIAN COORDINATES
C
      SUBROUTINE YPR(N,T,Y,YD)
      DIMENSION Y(4),YD(4)
      COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE
C
C  Enter the equations here
C
100   RMAG = SQRT(Y(1)**2 + Y(3)**2)
      QS = (CS*OMEGA)/2.0
      FB = (BK*(RMAG - G))/(RM*RMAG)
      FS = SK/RM
      IF (RMAG.LT.G) FB = 0.
      YD(1) = Y(2)
      YD(2) = -FB*Y(1) - FS*Y(1) - (CS*Y(2))/RM - (QS*Y(3))/RM +
     .EP*(OMEGA**2)*COS(OMEGA*T) + FSIDE/RM
      YD(3) = Y(4)
      YD(4) = -FB*Y(3) - FS*Y(3) - (CS*Y(4))/RM + (QS*Y(1))/RM +
     .(OMEGA**2)*EP*SIN(OMEGA*T)
      RETURN
      END
```

```
C       FILE NAME: PSD.FOR              29-SEP-83               APW
C
C       FUNCTION:  USES THE FFT ROUTINE TO COMPUTE THE PSD FOR THE WHIRL
C                  OF THE SIMPLE JEFFCOT ROTOR MODEL.
C
        DIMENSION DUMMY(4096),F(1200),Y(1200),Q1(1200),Q2(1200)
        TYPE*,'INPUT NO. OF DATA POINTS AND SAMPLING PERIOD:'
        ACCEPT*,NN,SP
        FRINK = 1.0/(FLOAT(NN)*SP)
        N = NN/2
        OPEN(UNIT=1,TYPE='OLD')
        OPEN(UNIT=2,TYPE='NEW')
        DO 100 I = 1,3
                ICOUNT = 0
                FR = 0
                GO TO(110,120,130)I
110             READ(1,31) (DUMMY(J),J=1,NN)
                GO TO 140
120             READ(1,32) (DUMMY(J),J=1,NN)
                GO TO 140
130             READ(1,33) (DUMMY(J),J=1,NN)
140             REWIND 1
                CALL FOUR1(DUMMY,N,1)
                DO 150 K = 1,N,2
                        DUMP = 2.0*SQRT(DUMMY(K)**2 + DUMMY(K+1)**2)
                        ICOUNT = ICOUNT + 1
                        GO TO(160,170,180)I
160                     Y(ICOUNT) = DUMP
                        F(ICOUNT) = FR
                        FR = FR + FRINK
                        GO TO 150
170                     Q1(ICOUNT) = DUMP
                        GO TO 150
180                     Q2(ICOUNT) = DUMP
150                     CONTINUE

100     CONTINUE
        WRITE(2,11)(F(I),Y(I),Q1(I),Q2(I),I=1,ICOUNT)
        CLOSE(UNIT=1)
        CLOSE(UNIT=2)
        OPEN(UNIT=3,TYPE='NEW')
        IC = 4
        WRITE(3,12)IC,ICOUNT
        WRITE(3,13) 'FREQ(HZ)'
        WRITE(3,13) 'POWERY'
        WRITE(3,13) 'POWERQ1'
        WRITE(3,13) 'POWERQ2'
        CLOSE(UNIT=3)
31      FORMAT(E15.8)
32      FORMAT(15X,E15.8)
33      FORMAT(30X,E15.8)
11      FORMAT(4E11.4)
12      FORMAT(2I5)
13      FORMAT(A8)
        END
```

```
CCCCCC
C  FILE NAME: PFFT.FOR
C
C  AUTHOR:  ANGIE WISEMAN
C
C  FUNCTION:  COMPUTES THE FAST-FOURIER TRANSFORM USING THE COOLEY-TUKEY
C             ALGORITHM.
C
C  INPUTS:      DATA(I)          I = A POWER OF 2 (i.e. 2**12)
C               NN               NN = I/2
C               ISIGN            = 1 FOR FFT; = -1 FOR [FFT]**-1
CCCCCC
        SUBROUTINE FOUR1 (DATA,NN,ISIGN)
        DIMENSION DATA(5000)
        INK=1
        N=2*NN
        J=1
        DO 50 I=1,N,2
             IF (I-J) 10,20,20
10           TEMPR=DATA(J)
             TEMPI=DATA(J+INK)
             DATA(J)=DATA(I)
             DATA(J+INK)=DATA(I+INK)
             DATA(I)=TEMPR
             DATA(I+INK)=TEMPI
20           M=N/2
30           IF (J-M) 50,50,40
40           J=J-M
             M=M/2
             IF (M-2) 50,30,30
50           J=J+M
        MMAX=2
60      IF (MMAX-N) 70,100,100
70      ISTEP=2*MMAX
        THETA=6.283185307/FLOAT(ISIGN*MMAX)
        SINTH=SIN(THETA/2.0)
        WSTPR=-2.0*SINTH*SINTH
        WSTPI=SIN(THETA)
        WR=1.0
        WI=0.0
        DO 90 M=1,MMAX,2
             DO 80 I=M,N,ISTEP
                J=I+MMAX
                TEMPR=WR*DATA(J)-WI*DATA(J+INK)
                TEMPI=WR*DATA(J+INK)+WI*DATA(J)
                DATA(J)=DATA(I)-TEMPR
                DATA(J+INK)=DATA(I+INK)-TEMPI
                DATA(I)=DATA(I)+TEMPR
80              DATA(I+INK)=DATA(I+INK)+TEMPI
             TEMPR=WR
             WR=WR*WSTPR-WI*WSTPI+WR
90           WI=WI*WSTPR+TEMPR*WSTPI+WI
        MMAX=ISTEP
        GO TO 60
100     RETURN
        END
```

```
C  FILE NAME:  PIPC.FOR          19-SEP-83        APW
C
C               POLAR COORDINATES VERSION FOR C TYPE MOTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       THIS PROGRAM SOLVES THE D.E.'S FOR THE SIMPLE ROTOR MODEL
C       AND ITERATES TO DETERMINE THE BOUNDARY CONDITIONS NECESSARY
C       FOR A PERIODIC SOLUTION TO THE EQUATIONS TO EXIST.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PIPC
        DIMENSION Y(4),YTOL(4),YO(4),GO(4),G1(4),GD(4,4),DELY(4)
        DIMENSION YOG(4),YDIFF(4),YH(4),YB(4),YOI(4)
      COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE,RSIDE,PHSIDE,CQ,QS
CCC
C    ACCEPT INPUT DATA FROM COMMAND FILE PITTER.COM:
CCC
        TYPE *,'INPUT THE INTEGRATION START AND STOP TIMES.'
        ACCEPT *,TSTART,TSTOP
        TYPE *,'INPUT THE TIME STEP AND NUMBER OF STEPS PER OUTPUT.'
        ACCEPT *,H,NPR
        IPR=NPR
        TYPE *,'INPUT THE DIMENSION OF THE STATE VECTOR Y.'
        ACCEPT *,N
        TYPE *,'INPUT THE INITIAL STATE VECTOR.'
        ACCEPT *,(Y(I),I=1,N)
        DO 2 I = 1,N
2       YOI(I) = Y(I)
        TYPE *,'INPUT THE VALUE OF DELTA:'
        ACCEPT *,DELTA
        TYPE *,'Enter values for the following quantities on separate'
        TYPE *,'lines: RM, BK, SK, CS, G, EP, OMEGA, FSIDE,& CQ'
        ACCEPT *,RM
        ACCEPT *,BK
        ACCEPT *,SK
        ACCEPT *,CS
        ACCEPT *,G
        ACCEPT *,EP
        ACCEPT *,OMEGA
        ACCEPT *,FSIDE
        ACCEPT *,CQ
CCC
C    COMPLETE THE INITIALIZATION
CCC
        QS = (CS*OMEGA)/(2.0*RM)
        FSIDE = FSIDE/RM
        HO6=H/6.
        HO2=H/2.
        T = TSTART
        OPEN (UNIT=1,TYPE='NEW')
```

```
      WRITE (1,*) 'Y(0)=',(Y(I),I=1,N)
      WRITE (1,*) '-------------------------------------------------
     &------------------------'
CCC
C
C   HERE WE GO..... TOP OF LOOP ................................
C
C     INITIALIZE YO
CCC
1       CONTINUE
        DO 100 K=1,N
100     YO(K) = Y(K)
CCC
C    INTEGRATE TO FIND Y(T)
CCC
110     CALL RUNK(N,T,Y,HO2,HO6,H)
        IF (T.LT.TSTOP) GO TO 110
- - ->  Y(3) = Y(3) - 6.283185308
CCC
C    COMPUTE G(YO) AND CHECK FOR CONVERGENCE
CCC
        DO 200 L = 1,N
200     GO(L) = Y(L) - YO(L)
        GMAG = SQRT(GO(1)**2 + GO(2)**2 + GO(3)**2 + GO(4)**2)
        DO 410 J = 1,N
        YDIFF(J) = ABS(GO(J))
410     YTOL(J) = ABS(0.01*YO(J))
        WRITE(1,*)'YDIF:',(YDIFF(I),I=1,N)
        WRITE(1,*)'YTOL:',(YTOL(I),I=1,N)
        WRITE(1,*)'  '
        WRITE(1,*)'GMAG:',GMAG
        WRITE(1,*)'  '
        IF((GMAG.GT.GMAGL).AND.(IZERO.EQ.1)) GO TO 500
        GMAGL = GMAG
        IZERO = 0
        IF(GMAG.GT.1.E-2) GO TO 400
        IF (YDIFF(1).GT.YTOL(1)) GO TO 400
        IF (YDIFF(3).LT.YTOL(3)) NNN = 1
        IF(NNN.EQ.1)GO TO 300
CCC
C    COMPUTE THE JACOBIAN OF G(Y)--->   J = (G(YO + DEL) - G(YO))/DEL
CCC
400     DO 2000 JI = 1,N
             DL = DELTA*YO(JI)
             IF (DL.GT.DELTA) DEL = DL
             IF (DELTA.GT.DL) DEL = DELTA
             YHOLD = YO(JI)
             YOG(JI) = YO(JI) + DEL
             DO 210 M = 1,N
                  IF (M.EQ.JI) GO TO 210
                  YOG(M) = YO(M)
210               Y(M) = YOG(M)
             T = TSTART
```

```
220          CALL RUNK(N,T,Y,H02,H06,H)
             IF (T.LT.TSTOP) GO TO 220
- - ->  Y(3) = Y(3) - 6.283185308
             DO 230 J = 1,N
                  G1(J) = Y(J) - YOG(J)
                  GD(J,JI) = (G1(J) - GO(J))/DEL
          IF(GD(J,JI).NE.0.) GO TO 230
                IZERO = 1
                IDX = JI
                IF(IDX.NE.4) IDX = 0
230           CONTINUE
             YO(JI) = YHOLD
2000    CONTINUE
CCC
C                                                    -1
C    COMPUTE THE VALUE OF DELY--->      DELY = [J]  * -G(YO)
C
C       INVERT THE JACOBIAN MATRIX:
CCC
        WRITE(1,*)'THE JACOBIAN:'
        CALL MRITE(GD,N,N,0,7)
        CALL MRITE(GD,N,N,0,1)
        CALL MINV(GD,N,1.E-21,DET,NPIV)
        WRITE(1,*)'INVERSE JACOBIAN:'
        CALL MRITE(GD,N,N,0,1)
CCC
C  IF THE PIVOT VALUE IS LESS THAN E, WHAT RETURNS FROM MINV IS GARBAGE.
C  USE THE FOLLOWING TO COMPUTE A DELTA & TRY AGAIN.
CCC
        IF (NPIV.GE.3) GO TO 330
        IF (NPIV.EQ.0) GO TO 241
        DO 247 L = 1,N
247     DELY(L) = DELTA*100.0*YO(L)
        GO TO 242
241     DO 240 K = 1,N
240     GO(K) = -GO(K)
CCC
C       PERFORM VECTOR MATRIX PRODUCT:
CCC
        CALL MVMU(GD,GO,DELY,N,N)
CCC
C    COMPUTE NEW VALUE OF THE VECTOR Y AND REPEAT
CCC
242     DO 250 M = 1,N
        Y(M) = YO(M) + DELY(M)
250     YB(M) = Y(M)
        IF(Y(1).LT.0.)Y(1)=YOI(1)
        WRITE(1,*)'DELY(I)=',(DELY(I),I=1,N)
        WRITE(1,*)'  '
        WRITE(1,*) 'Y(I)=',(Y(I),I=1,N)
        WRITE(1,*)'  '
        T = TSTART
        ITTER = ITTER +1
        IF (ITTER.LE.20) GO TO 1
```

```
      GO TO 310
500   IZERO = 0
      WRITE(1,*)'AT 500, RESETTING Y TO ICS EXCEPT FOR IDX COMPONENT'
      CHG = YB(IDX)*0.20
      DO 510 LL = 1,N
510   Y(LL) = YOI(LL)
      Y(IDX) = CHG
      WRITE(1,*)'NEW Y:',(Y(I),I=1,N)
      ITTER = ITTER + 1
      GO TO 1
300   WRITE(1,*)'TOLERANCE MET, ITERATIONS:',ITTER
      WRITE(1,*)'FINAL Y(I)=',(YO(I),I=1,N)
      GO TO 320
310   WRITE(1,*)'MAX NO. OF ITTERATIONS EXCEEDED;  GMAG =',GMAG
      GO TO 320
330   WRITE(1,*)'PIVOT VALUE LESS THAN E THREE TIMES, PROGRAM STOPPED.'
320   CLOSE(UNIT=1)
      STOP
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME: PRK4.FOR              15-NOV-83                APW
C
C       THIS FILE CONTAINS A RUNGE-KUTTA INTEGRATION ROUTINE WHICH
C       ACCOMPANIES THE MAIN PROGRAM IN PIP.FOR AND PIPC.FOR
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SUBROUTINE LISTING:
C
C       RUNK (A RUNGE-KUTTA INTEGRATION ROUTINE)        @A
C       YPR  (CONTAINS THE SYSTEM STATE EQUATIONS)      @B
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       RUNGE-KUTTA INTEGRATION LOOP DRIVER.
C       THIS PROGRAM ASSUMES THE DERIVATIVE FUNCTION YPR(T,Y) IS
C       DEFINED EXTERNALLY BY A SUBROUTINE CALLED YPR.  THE DATA
C       IS ASSUMED TO BE PASSED THROUGH THE CALL STATEMENT IN THE
C       FORM CALL YPR(N,T,Y,YD); WHERE N IS THE DIMENSION OF THE
C       STATE VECTOR, T IS THE INDEPENDENT VARIABLE (TIME), Y IS
C       THE STATE VECTOR AND YD IS THE DERIVATIVE OF THE STATE
C       VECTOR.  INPUTS TO THE PROGRAM ARE FROM THE TERMINAL OR
C       A COMMAND FILE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C@A
        SUBROUTINE RUNK(N,T,Y,HO2,HO6,H)
      DIMENSION Y(4),YD(4),YDO(4),YD1(4),YD2(4),YD3(4),TEMP(4)
        CALL YPR(N,T,Y,YDO)
        T=T+HO2
        DO 1000 I=1,N
1000    TEMP(I)=Y(I)+HO2*YDO(I)
        CALL YPR(N,T,TEMP,YD1)
        DO 1010 I=1,N
1010    TEMP(I)=Y(I)+HO2*YD1(I)
        CALL YPR(N,T,TEMP,YD2)
        T=T+HO2
        DO 1020 I=1,N
1020    TEMP(I)=Y(I)+H*YD2(I)
        CALL YPR(N,T,TEMP,YD3)
        DO 1030 I=1,N
1030    Y(I)=Y(I)+HO6*(YDO(I)+YD3(I)+2*(YD1(I)+YD2(I)))
        RETURN
        END

CCC
C  The equations given below describe the SIMPLE SSME deadband rotor model...
C@B
      SUBROUTINE YPR(N,T,Y,YD)
      DIMENSION Y(4),YD(4)
      COMMON / PASS / RM,BK,SK,CS,G,EP,OMEGA,FSIDE,RSIDE,PHSIDE,CQ,QS
      FB = - BK*(Y(1) - G)/RM
      FS = - SK*Y(1)/RM
      YNL2 = Y(1)*(Y(4)**2)
```

```
YNL4 = - (2.0*Y(2)*Y(4))/Y(1)
CB = EP*(OMEGA**2)*COS(OMEGA*T - Y(3))
SB = ((OMEGA**2)*EP*SIN(OMEGA*T - Y(3)))/Y(1)
RSIDE = FSIDE*COS(Y(3))
PHSIDE = -(FSIDE*SIN(Y(3)))/Y(1)
CS2 = - (CS*Y(2))/RM
CS4 = - (CS*Y(4))/RM
CQ2 = - (CQ*Y(1)*Y(4))/RM
CQ4 =   (CQ*Y(2))/(RM*Y(1))
IF (Y(1).LT.G) FB = 0.
YD(1) = Y(2)
YD(2) =  FB + FS + CS2 + CQ2 + YNL2 + CB + RSIDE
YD(3) = Y(4)
YD(4) = CS4 + CQ4 + QS + YNL4 + SB + PHSIDE
RETURN
END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME: PMTRX.FOR            15-NOV-83               APW
C
C       THIS FILE CONTAINS THE MATRIX SUBROUTINES WHICH ACCOMPANY
C       THE MAIN PROGRAM IN PIP.FOR AND PIPC.FOR
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SUBROUTINE LISTING:
C
C       MINV (A MATRIX INVERSE ROUTINE)                 @1
C       MVMU (MULTIPLIES A MATRIX AND A VECTOR)         @2
C       MRITE (WRITES THE MATRIX)                       @3
C
C@1
        SUBROUTINE MINV(A,N,E,DET,NPIV)
        DIMENSION A(N,N),IROW(50),JCOL(50),JORD(50),Y(50)
        M=N
        IF (N.LE.50) GOTO 5
        TYPE *,'DIMENSION OF MATRIX > 50.'
        RETURN
5       DET=1
        DO 18 K=1,N
        KM1=K-1
        PIVOT=0.0
        DO 11 I=1,N
        DO 11 J=1,N
        IF (K.EQ.1) GOTO 9
        DO 8 ISCAN=1,KM1
        DO 8 JSCAN=1,KM1
        IF (I.EQ.IROW(ISCAN)) GOTO 11
        IF (J.EQ.JCOL(JSCAN)) GOTO 11
8       CONTINUE
9       IF (ABS(A(I,J)).LE.ABS(PIVOT)) GOTO 11
        PIVOT=A(I,J)
        IROW(K)=I
        JCOL(K)=J
11      CONTINUE
        IF (ABS(PIVOT).GT.E) GOTO 13
        NPIV = NPIV + 1
        NPL = NPIV
        TYPE*,'PIVOT VALUE LESS THAN E.'
        RETURN
13      IROWK=IROW(K)
        JCOLK=JCOL(K)
        DET=DET*PIVOT
        IF (NPIV.EQ.NPL) NPIV = 0
        DO 14 J=1,M
14      A(IROWK,J)=A(IROWK,J)/PIVOT
        A(IROWK,JCOLK)=1./PIVOT
        DO 18 I=1,N
        AIJCK=A(I,JCOLK)
        IF (I.EQ.IROWK) GOTO 18
        A(I,JCOLK)=-AIJCK/PIVOT
```

```
        DO 17 J=1,M
17      IF (J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
18      CONTINUE
        DO 20 I=1,N
        IROWI=IROW(I)
        JCOLI=JCOL(I)
        JORD(IROWI)=JCOLI
20      CONTINUE
        INTCH = 0
        NM1=N-1
        DO 22 I=1,NM1
        IP1=I+1
        DO 22 J=IP1,N
        IF (JORD(J).GE.JORD(I)) GOTO 22
        JTEMP = JORD(J)
        JORD(J)=JORD(I)
        JORD(I)=JTEMP
        INTCH = INTCH + 1
22      CONTINUE
        IF (INTCH/2*2.NE.INTCH) DET=-DET
26      DO 28 J=1,N
        DO 27 I=1,N
        IROWI=IROW(I)
        JCOLI=JCOL(I)
27      Y(JCOLI)=A(IROWI,J)
        DO 28 I=1,N
28      A(I,J)=Y(I)
        DO 30 I=1,N
        DO 29 J=1,N
        IROWJ=IROW(J)
        JCOLJ=JCOL(J)
29      Y(IROWJ)=A(I,JCOLJ)
        DO 30 J=1,N
30      A(I,J)=Y(J)
        RETURN
        END
CCC@2
C   MVMU MULTIPLIES A MATRIX BY A VECTOR
C
C   ****** T(M) = A(M,N) * U(N) ******
CCC
      SUBROUTINE MVMU ( A, U, T, M, N )
      DIMENSION U ( N ), T ( M ), A ( M, N )
       I = 1
20051 if (.not.( I .le. M)) goto 20053
      T ( I ) = 0
20052 I = I + 1
      goto 20051
20053 continue
       I = 1
20054 if (.not.( I .le. M)) goto 20056
      continue
       J = 1
20057 if (.not.( J .le. N)) goto 20059
      T ( I ) = A ( I, J ) * U ( J ) + T ( I )
```

```
20058 J = J + 1
      goto 20057
20059 continue
20055 I = I + 1
      goto 20054
20056 continue
      RETURN
      END

CCC@3
C
C  MRITE WRITES A MATRIX OF UP TO 30 X 30 IN REASSEMBLABLE FORM
C
CCC
      SUBROUTINE MRITE(MAT,M,N,LFORM,IX)
      REAL*4 MAT(M,N)
      IF (N.GT.10) GOTO 11
      DO 1 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=1,N)
 1    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=1,N)
CC      WRITE (IX,208)
      GOTO 140
 11   IF ( N.GT.20 ) GOTO 110
      DO 2 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=1,10)
 2    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=1,10)
CC      WRITE (IX,208)
      DO 3 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=11,N)
 3    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=11,N)
      GOTO 140
 110  IF ( N.GT.30 ) GOTO 130
      DO 4 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=1,10)
 4    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=1,10)
CC      WRITE (IX,208)
      DO 5 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=11,20)
 5    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=11,20)
CC      WRITE (IX,208)
      DO 6 II=1,N
      I=II
      IF (LFORM.EQ.1) WRITE (IX,201) (MAT(I,J),J=21,N)
 6    IF (LFORM.NE.1) WRITE (IX,200) (MAT(I,J),J=21,N)
CC      WRITE (IX,208)
      GOTO 140
 130  IF (N.GT.30) WRITE (IX,300)
```

```
 140  CONTINUE
 200  FORMAT (10E13.5)
 201  FORMAT (10F12.5)
CC 208  FORMAT (' END OF LIST')
 300  FORMAT (' ARRAY DIMENSION >30')
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PQUAD.FOR           17-NOV-83       APW
C
C       FUNCTION:  COMPUTES THE COEF'S OF THE SECOND ORDER
C                  CHARACTERISTIC EQUATION FOR THE ROTOR SYSTEM
C                  AND DETERMINES THE ROOTS THEREOF.
C                       FS=G=EP=0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PQUAD
        DATA RM,CS,CQ,TK,QS,QSMAX/.20422,200.,40.,1.2E6,484810.,484824./
        OPEN(UNIT=1,TYPE='NEW',NAME='PQUAD.DAT')
        C1 = CS/RM
        WRITE(1,*)'FOR ALL CASES, C1 =  ',C1
        WRITE(1,*)'    '
1       C2 = (TK*(CS**2) + CQ*(CS**2) - (QS**2)*RM)/((CS**2)*RM)
        TYPE*,'QS,C1, & C2:',QS,C1,C2
        WRITE(1,*)'QS:',QS,'C2:',C2
        WRITE(1,*)'    '
        R1 = -C1/2.
        RAD = C1**2 - 4.0*C2
        IF(RAD.GT.0.) GO TO 10
        RAD = - RAD
        RR1 = R1
        RR2 = R1
        RI1 = SQRT(RAD)/2.0
        RI2 = -RI1
        GO TO 20
10      RI1 = 0.
        RI2 = 0.
        RR1 = R1 + (SQRT(RAD))/2.0
        RR2 = R1 - (SQRT(RAD))/2.0
20      WRITE(1,*)'ROOT1:',RR1,RI1
        WRITE(1,*)'ROOT2:',RR2,RI2
        WRITE(1,*)'    '
        WRITE(1,*)'    '
        QS = QS + .5
        IF(QS.GT.QSMAX)GO TO 100
        GO TO 1
100     TYPE*,'QSMAX EXCEEDED, PROGRAM STOPPED'
        WRITE(1,*)'QSMAX EXCEEDED, PROGRAM STOPPED'
        CLOSE(UNIT=1)
        STOP
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  FILE NAME: PQ4.FOR   18-NOV-83                APW
C
C  FUNCTION:  COMPUTES THE EQUILIBRIUM WHIRL ORBIT RADIUS, THEN
C             COMPUTES THE CEOFFICIENTS OF THE CHARACTERISTIC EQUATION
C             FOR THE ROTOR SYSTEM WITH G NOT EQUAL TO ZERO, SIDE
C             FORCES AND ROTOR ECCENTRICITY ARE ZERO.
C             (FOURTH ORDER LINEARIZED MODEL)
C               EP = FS = 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        DIMENSION G(5)
        IMPLICIT REAL*8(A-H,O-Z)
        OPEN(UNIT=1,TYPE='NEW',NAME='PQ4.DAT')
        CALL HORIZ(16,1)
        TYPE*,'INPUT QSMIN,QSMAX,QSINC:'
        ACCEPT*,QSMIN,QSMAX,QSINC
        BK = 1.0D6
        SK = 2.0D5
        CS = 200.000
        CQ = 40.00
        RM = 0.2042200
        G(1) = 0.5D-3
        G(2) = 1.0D-3
        G(3) = 1.5D-3
        G(4) = 2.0D-3
        G(5) = 2.5D-3
        PI = 3.141592654
        QS = QSMIN
        OM = QS/100.
CCC
C  COMPUTE THE COEFFICENTS A3,A2,A1,AO
C  THE C.E. HAS THE FORM:  X**4 + A3*X**3 + A2*X**2 + A1*X + AO
CCC
        DO 1000 I = 1,5
        J = I
        TK = BK + SK
        TYPE 21,J,G(I)
        WRITE(1,21)J,G(I)
1       TYPE 20,QS
        WRITE(1,20) QS
CCC
C  COMPUTE THE RADIUS OF THE WHIRL ORBIT
CCC
        RO = (BK*(CS**2)*G(I))/((CS**2)*TK + CQ*QS*CS - (QS**2)*RM)
        TYPE*,'RO:',RO
        WRITE(1,*)'RO:',RO
        A3 = (2.0*CS)/RM
        A2 = ((CS**2)+(CQ**2))/(RM**2) + (TK*(RO + 1.0))/RM
        A1 = ((QS*CQ + CS*TK)*RO + (QS*CQ + CS*TK))/(RM**2)
        AO = (RO*(QS**2 + TK**2))/(RM**2)
          WRITE(1,*)'    '
          WRITE(1,*)'A3',A3,'A2:',A2,'A1:',A1,'AO:',AO
```

```
        QS = QS + QSINC
        OM = QS/100.
        IF (QS.GT.QSMAX) GO TO 200
        GO TO 1
200     TYPE*,'QSMAX EXCEEDED, G INCREMENTED'
        TYPE*,'  '
        WRITE(1,*)'QSMAX EXCEEDED, G INCREMENTED'
        WRITE(1,*)'    '
        WRITE(1,*)'    '
1000    QS = QSMIN
20      FORMAT('QS:',E15.8)
21      FORMAT('G(',I1,'):',F15.6)
        CALL HORIZ(10,1)
        CLOSE(UNIT=1)
        END

      SUBROUTINE HORIZ(I,N)
      IF ( I .EQ. 10 ) GOTO 100
      IF ( I .EQ. 16 ) GOTO 130
 10   FORMAT(10A1)
 100  WRITE(N,10) 27,91,49,119
      GOTO 200
 130  WRITE(N,10) 27,91,52,119
      GOTO 200
 200  CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  FILE NAME: PQC.FOR   16-NOV-83                APW
C
C  FUNCTION:  COMPUTES THE CEOFFICIENTS OF THE CHARACTERISTIC EQUATION
C             FOR THE ROTOR SYSTEM WITH G NOT EQUAL TO ZERO, SIDE
C             FORCES AND ROTOR ECCENTRICITY ARE ZERO.
C             (THIRD ORDER SYSTEM)
C               EP = FS = 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        DATA BK,SK,CS,CQ,RM/1.0E6,2.0E5,200.,40.,0.20422/
        DATA QS,QSMAX/504792.1000,504792.2000/
        DATA PI/3.141592654/
        OPEN(UNIT=1,TYPE='NEW',NAME='PQROOT.DAT')
CCC
C  COMPUTE THE COEFFICENTS P,Q,R
C  THE C.E. HAS THE FORM:  Y**3 + P*Y**2 + Q*Y + R = 0
CCC
1       TYPE 20,QS
        WRITE(1,20) QS
        P = (2.0*CS)/RM
        TK = BK + SK
        PHIO = QS/CS
        Q=(CS**2+CQ**2+RM*(TK+CQ*PHIO-RM*PHIO**2-2.*PHIO*CQ))/RM**2
        R=(CS*(TK + CQ*PHIO - RM*PHIO**2))/RM**2
        QS = QS + 10.
        WRITE(1,10) P,Q,R
        TYPE 10,P,Q,R
        IF (QS.GT.QSMAX) GO TO 200
        GO TO 1
200     TYPE*,'QSMAX EXCEEDED'
        WRITE(1,*)'QSMAX EXCEEDED'
10      FORMAT(2X,'P,Q,R:',3(2X,F15.6))
20      FORMAT(2X,'QS:',F15.6)
        CLOSE(UNIT=1)
        STOP
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PREP.FOR            14-DEC-83       APW
C
C       FUNCTION:  COMPUTES THE COEF'S OF THE SECOND ORDER EQUATION
C                  USED TO DETERMINE THE EQUILIBRIUM RADIUS OF THE
C                  ROTOR FOR THE CASES WHERE EP AND G ARE NOT ZERO WITH
C                  WITH FS ZERO.  THE QUADRATIC IS THEN SOLVED TO DETER-
C                  MINE THE RADIUS.  FOR PLOTTING!!
C                       FS = 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PREP
        DIMENSION GG(9),OMP(1000),RMG(1000,9)
        BK = 1.0E6
        SK = 2.0E5
        TK = BK + SK
        CS = 200.
        CQ = 40.
        RM = 0.20422
        TYPE*,'CHOOSE A SPECIFIC DEADBAND & OFFSET---'
        TYPE*,'INPUT NUMBER OF DEADBANDS (UP TO 9)'
        ACCEPT*,NN
        TYPE*,'INPUT DEADBAND VALUES'
        ACCEPT*,(GG(M),M=1,NN)
        TYPE*,'INPUT THE OFFSET VALUE '
        ACCEPT*,EP
200     TYPE *,'INPUT QSMIN & QSMAX,QSINC:'
        ACCEPT*,QSMIN,QSMAX,QSINC
        QS = QSMIN
        DO 1000 I = 1,NN
        G = GG(I)
        J = 1
1       OM = QS/100.
        A = (BK + SK + CQ*OM - RM*OM**2)
        B = (QS - CS*OM)
        AA = A**2 + B**2
        BB = G*(RM*A*(OM**2) - B**2)
        C=(EP**2)*(RM**2)*(OM**4)*(A**2+B**2) - (BK**2)*(G**2)*(B**2)
        IF (C.GE.0) GO TO 35
        RADIUS = 0.
        GO TO 50
35      ROOT1 = G + (BB + SQRT(C))/AA
        ROOT2 = G + (BB - SQRT(C))/AA
        IF((ROOT1.GE.0.).OR.(ROOT2.GE.0.))GO TO 30
        RADIUS = 0.
        GO TO 50
30      IF((ROOT1.GT.0.).AND.(ROOT1.GT.ROOT2))RADIUS = ROOT1
        IF((ROOT2.GT.0.).AND.(ROOT2.GT.ROOT1))RADIUS = ROOT2
CC      RMG(J,I) = RADIUS - G
        RMG(J,I) = RADIUS
50      IF(RMG(J,I).LT.0.) RMG(J,I) = 0.
        OMP(J) = OM
        J = J + 1
        QS = QS + QSINC
```

```
      IF(QS.GT.QSMAX)GO TO 100
      GO TO 1
100   QS = QSMIN
1000  CONTINUE
      ICOUNT = J - 1
      OPEN(UNIT=1,TYPE='NEW',NAME='PREP1.DAT')
      KK= NN + 1
      GO TO (22,22,33,44,55,66,77,88,99,111),KK
22    WRITE(1,2)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
33    WRITE(1,3)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
44    WRITE(1,4)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
55    WRITE(1,5)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
66    WRITE(1,6)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
77    WRITE(1,7)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
88    WRITE(1,8)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
99    WRITE(1,9)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
      GO TO 9999
111   WRITE(1,10)(OMP(L),(RMG(L,K),K=1,NN),L=1,ICOUNT)
9999  CLOSE(UNIT=1)
      OPEN(UNIT=2,TYPE='NEW',NAME='PREP2.DAT')
      ICOL = KK
      WRITE(2,11)ICOL,ICOUNT
      WRITE(2,12)'OMEGA'
      WRITE(2,12)'G1'
      WRITE(2,12)'G2'
      WRITE(2,12)'G3'
      WRITE(2,12)'G4'
      WRITE(2,12)'G5'
      WRITE(2,12)'G6'
      WRITE(2,12)'G7'
      WRITE(2,12)'G8'
      WRITE(2,12)'G9'
      CLOSE(UNIT=2)
2     FORMAT(2E11.4)
3     FORMAT(3E11.4)
4     FORMAT(4E11.4)
5     FORMAT(5E11.4)
6     FORMAT(6E11.4)
7     FORMAT(7E11.4)
8     FORMAT(8E11.4)
9     FORMAT(9E11.4)
10    FORMAT(10E11.4)
11    FORMAT(2I5)
12    FORMAT(A8)
      STOP
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PREPBF.FOR          14-DEC-83       APW
C
C       FUNCTION:  COMPUTES THE COEF'S OF THE SECOND ORDER EQUATION
C                  USED TO DETERMINE THE EQUILIBRIUM RADIUS OF THE
C                  ROTOR FOR THE CASES WHERE EP AND G ARE NOT ZERO WITH
C                  WITH FS ZERO.  THEN COMPUTES THE RADIUS AND THE
C                  RESULTING BEARING FORCE.  FOR PLOTTING!
C                       FS = 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PREP
        DIMENSION GG(9),OMP(1000),BF(1000,9)
        BK = 1.0E6
        SK = 2.0E5
        TK = BK + SK
        CS = 200.
        CQ = 40.
        RM = 0.20422
        TYPE*,'CHOOSE A SPECIFIC DEADBAND & OFFSET---'
        TYPE*,'INPUT NUMBER OF DEADBANDS (UP TO 9)'
        ACCEPT*,NN
        TYPE*,'INPUT DEADBAND VALUES'
        ACCEPT*,(GG(M),M=1,NN)
        TYPE*,'INPUT THE OFFSET VALUE '
        ACCEPT*,EP
200     TYPE *,'INPUT QSMIN & QSMAX,QSINC:'
        ACCEPT*,QSMIN,QSMAX,QSINC
        QS = QSMIN
        DO 1000 I = 1,NN
        G = GG(I)
        J = 1
1       OM = QS/100.
        A = (BK + SK + CQ*OM - RM*OM**2)
        B = (QS - CS*OM)
        AA = A**2 + B**2
        BB = G*(RM*A*(OM**2) - B**2)
        C=(EP**2)*(RM**2)*(OM**4)*(A**2+B**2) - (BK**2)*(G**2)*(B**2)
        IF (C.GE.0) GO TO 35
        RADIUS = 0.
        GO TO 50
35      ROOT1 = G + (BB + SQRT(C))/AA
        ROOT2 = G + (BB - SQRT(C))/AA
        IF((ROOT1.GE.0.).OR.(ROOT2.GE.0.))GO TO 30
        RADIUS = 0.
        GO TO 50
30      IF((ROOT1.GT.0.).AND.(ROOT1.GT.ROOT2))RADIUS = ROOT1
        IF((ROOT2.GT.0.).AND.(ROOT2.GT.ROOT1))RADIUS = ROOT2
50      RMG = RADIUS - G
        RTY = RMG
        IF(RMG.LT.0.) RMG = 0.
        BF(J,I) = BK*RMG
        BTY = BF(J,I)
```

```
        OMP(J) = OM
        QS = QS + QSINC
        IF(QS.GT.QSMAX)GO TO 100
        J = J + 1
        GO TO 1
100     QS = QSMIN
1000    CONTINUE
        ICOUNT = J
        OPEN(UNIT=1,TYPE='NEW',NAME='PREP1.DAT')
        KK= NN + 1
        GO TO (22,22,33,44,55,66,77,88,99,111),KK
22      WRITE(1,2)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
33      WRITE(1,3)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
44      WRITE(1,4)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
55      WRITE(1,5)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
66      WRITE(1,6)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
77      WRITE(1,7)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
88      WRITE(1,8)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
99      WRITE(1,9)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
        GO TO 9999
111     WRITE(1,10)(OMP(L),(BF(L,K),K=1,NN),L=1,ICOUNT)
9999    CLOSE(UNIT=1)
        OPEN(UNIT=2,TYPE='NEW',NAME='PREP2.DAT')
        ICOL = KK
        WRITE(2,11)ICOL,ICOUNT
        WRITE(2,12)'OMEGA'
        WRITE(2,12)'G1'
        WRITE(2,12)'G2'
        WRITE(2,12)'G3'
        WRITE(2,12)'G4'
        WRITE(2,12)'G5'
        WRITE(2,12)'G6'
        WRITE(2,12)'G7'
        WRITE(2,12)'G8'
        WRITE(2,12)'G9'
        CLOSE(UNIT=2)
2       FORMAT(2E11.4)
3       FORMAT(3E11.4)
4       FORMAT(4E11.4)
5       FORMAT(5E11.4)
6       FORMAT(6E11.4)
7       FORMAT(7E11.4)
8       FORMAT(8E11.4)
9       FORMAT(9E11.4)
10      FORMAT(10E11.4)
11      FORMAT(2I5)
12      FORMAT(A8)
        STOP
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PSRO.FOR            22-NOV-83       APW
C
C       FUNCTION:  COMPUTES RO FOR VARIOUS SIDE FORCES, FREQUENCIES,
C                  AND DEADBANDS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PSRO
        DIMENSION G(6),GI(6),JDB(6)
        DATA GI /0.5E-3,1.0E-3,1.5E-3,2.0E-3,2.5E-3,0.0/
        BKI = 1.0E6
        SK = 2.0E5
        CSI = 200.
        CQ = 40.
        RM = 0.20422
        BK = BKI
        CS = CSI
        TYPE*,'INPUT THE NUMBER OF DEADBAND VALUES TO BE CONSIDERED,'
        TYPE*,'ONE TO SIX:'
        ACCEPT*,NDB
        IF(NDB.LT.6) GO TO 2
        DO 300 L = 1,6
300     G(L) = GI(L)
        GO TO 3
2       TYPE*,'INPUT THE SPECIFIC DEADBANDS DESIRED:'
        TYPE*,'(1).5E-3,(2)1.E-3,(3)1.5E-3,(4)2.E-3,(5)2.5E-3,(6)0.0'
        ACCEPT*,(JDB(K),K=1,NDB)
        DO 700 I = 1,NDB
        JI = JDB(I)
700     G(I) = GI(JI)
3       TYPE*,'INPUT FSI,FSMAX, & FSINC:'
        ACCEPT*,FSI,FSMAX,FSINC
        TYPE *,'INPUT QSMIN,QSMAX, & QSINCI:'
        ACCEPT*,QSMIN,QSMAX,QSINCI
        QS = QSMIN
        FS = FSI
        QSINC = QSINCI
        OPEN(UNIT=1,TYPE='NEW',NAME='PSRO.DAT')
        CALL HORIZ(16,1)
        DO 1000 K = 1,NDB
4       WRITE(1,*)'G:',G(K),'    FS:',FS
        WRITE(1,*)'   '
1       FSMIN = SQRT((SK**2 + QS**2)*G(K)**2)
        IF(FS.LT.FSMIN) BK = 0.
        OM = QS/100.
        IF(QS.EQ.0.) CS = 0.
           A = (BK + SK)**2 + QS**2
           B = (BK + SK)*BK*G(K)
           C = FS**2*A - QS**2*BK**2*G(K)**2
           R1 = B/A + SQRT(C)/A
           R2 = B/A - SQRT(C)/A
           STH1 = (R1*QS)/FS
           STH2 = (R2*QS)/FS
```

```
            ARG1 = STH1/(SQRT(1.0 - STH1**2))
            ARG2 = STH2/(SQRT(1.0 - STH2**2))
            TH1 = ATAN(ARG1)
            TH2 = ATAN(ARG2)
            WRITE(1,*)'OM:',OM,'  FS:',FS,'  FSMIN:',FSMIN
            WRITE(1,*)'  R1:',R1,'    R2:',R2
            WRITE(1,*)' TH1:',TH1,'   TH2:',TH2
            WRITE(1,*)'   '
            BK = BKI
            GO TO 40
40          QS = QS + QSINC
            IF(QS.GT.0.) CS = CSI
            IF(QS.GT.QSMAX) GO TO 100
            GO TO 1
100          IF(FS.GE.FSMAX) GO TO 400
             FS = FS + FSINC
             QS = QSMIN
             QSINC = QSINCI
            GO TO 4
400          FS = FSI
             QS = QSMIN
             QSINC = QSINCI
1000     CONTINUE
         TYPE*,'FINISHED'
         CALL HORIZ(10,1)
         CLOSE(UNIT=1)
         STOP
         END

      SUBROUTINE HORIZ(I,N)
      IF ( I .EQ. 10 ) GOTO 100
      IF ( I .EQ. 16 ) GOTO 130
 10   FORMAT(10A1)
 100  WRITE(N,10) 27,91,49,119
      GOTO 200
 130  WRITE(N,10) 27,91,52,119
      GOTO 200
 200  CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PSROOT.FOR          08-DEC-83       APW
C
C       FUNCTION:  COMPUTES RO FOR VARIOUS SIDE FORCES, FREQUENCIES,
C                  AND DEADBANDS.  THE VALUE OF RO IS THEN USED TO
C                  DETERMINE THE ROOTS OF THE CHARACTERISTIC EQUATION
C                  WHICH HAS BEEN CAST IN THE FORM OF A PRODUCT OF
C                  TWO SECOND ORDER EQUATIONS IN 'S'.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PSROOT
        DIMENSION G(6),GI(6),JDB(6)
        DATA GI /0.5E-3,1.0E-3,1.5E-3,2.0E-3,2.5E-3,0.0/
        BKI = 1.0E6
        SK = 2.0E5
        CSI = 200.
        CQ = 40.
        RM = 0.20422
        BK = BKI
        CS = CSI
        TYPE*,'INPUT THE NUMBER OF DEADBAND VALUES TO BE CONSIDERED,'
        TYPE*,'ONE TO SIX:'
        ACCEPT*,NDB
        IF(NDB.LT.6) GO TO 2
        DO 300 L = 1,6
300     G(L) = GI(L)
        GO TO 3
2       TYPE*,'INPUT THE SPECIFIC DEADBANDS DESIRED:'
        TYPE*,'(1).5E-3,(2)1.E-3,(3)1.5E-3,(4)2.E-3,(5)2.5E-3,(6)0.0'
        ACCEPT*,(JDB(K),K=1,NDB)
        DO 701 I = 1,NDB
        JI = JDB(I)
701     G(I) = GI(JI)
3       TYPE*,'INPUT FSI,FSMAX, & FSINC:'
        ACCEPT*,FSI,FSMAX,FSINC
        TYPE *,'INPUT QSMIN,QSMAX, & QSINCI:'
        ACCEPT*,QSMIN,QSMAX,QSINCI
        TYPE*,'INPUT A "1" FOR OUTPUT TO BE PLOTTED.'
        ACCEPT*,IPLOT
        QS = QSMIN
        FS = FSI
        QSINC = QSINCI
        OPEN(UNIT=1,TYPE='NEW')
        IF(IPLOT.NE.1) GO TO 5
        OPEN(UNIT=2,TYPE='NEW')
        OPEN(UNIT=3,TYPE='NEW')
5       CALL HORIZ(16,1)
        DO 1000 K = 1,NDB
4       WRITE(1,*)'G:',G(K),'    FS:',FS
        WRITE(1,*)'   '
1       FSMIN = SQRT((SK**2 + QS**2)*G(K)**2)
```

```
        IF(FSMIN.GT.FS) BK=0.
        OM = QS/100.
        IF(QS.EQ.0.) CS = 0.
           A = (BK + SK)**2 + QS**2
           B = (BK + SK)*BK*G(K)
           C = FS**2*A - QS**2*BK**2*G(K)**2
           R1 = B/A + SQRT(C)/A
           R2 = B/A - SQRT(C)/A
CCC
C   COMPUTE THE ROOTS OF THE CHARACTERISTIC EQUATION
CCC
C
C  DETERMINE RO
CCC
        IF(BK.EQ.0) GO TO 520
        IF(R1.GE.G(K))RO = R1
        IF(R2.GE.G(K))RO = R2
        GO TO 540
520     IF(R1.GT.0.) RO = R1
        IF(R2.GT.0.) RO = R2
CCC
C   COMPUTE DELTA -- IF DELTA IS COMPLEX, RO IS COMPLEX
CCC
540     DELTA = (BK*G(K))/(2.0*RM*RO)
        RK = (SK+BK)/RM - DELTA
CCC
C  COMPUTE THE DELTA-Q RADICAL
CCC
        RAD = DELTA**2 - (QS**2/RM**2)
        IF(RAD.LT.0.)GO TO 530
        DQREAL = SQRT(RAD)
        DQIMAG = 0.
        GO TO 550
530     DQIMAG = SQRT(ABS(RAD))
        DQREAL = 0.
550     XKRP = (RK + DQREAL)*4.0
        XKIP =  DQIMAG*4.0
        XKRN = (RK - DQREAL)*4.0
        XKIN = -XKIP
        REALP = (CS**2/RM**2) - XKRP
        REALN = (CS**2/RM**2) - XKRN
        IF(XKIP.NE.0.) GO TO 560
        IF(REALP.GE.0) GO TO 561
        XR1 = 0.
        XI1 = SQRT(ABS(REALP))
        GO TO 562
561     XR1 = SQRT(REALP)
        XI1 = 0.
562     IF(REALN.GE.0) GO TO 563
        XR2 = 0.
        XI2 = SQRT(ABS(REALN))
        GO TO 565
563     XR2 = SQRT(REALN)
        XI2 = 0.
```

```
      GO TO 565
560   SQMP = SQRT(SQRT(REALP**2 + XKIP**2))/2.0
      SQMN = SQRT(SQRT(REALN**2 + XKIN**2))/2.0
      SQMPT = (ATAN2(XKIP,REALP))/2.0
      SQMNT = (ATAN2(XKIN,REALN))/2.0
      XR1 = SQMP*COS(SQMPT)
      XI1 = SQMP*SIN(SQMPT)
      XR2 = SQMN*COS(SQMNT)
      XI2 = SQMN*SIN(SQMNT)
565   RT1R = - CS/(2.0*RM) + XR1
      RT1I = XI1
      RT2R = - CS/(2.0*RM) - XR1
      RT2I = -XI1
      RT3R = - CS/(2.0*RM) + XR2
      RT3I = XI2
      RT4R = - CS/(2.0*RM) - XR2
      RT4I = -XI2
      IF(RT1R.GT.0) GO TO 700
      IF(RT2R.GT.0) GO TO 700
      IF(RT3R.GT.0) GO TO 700
      IF(RT4R.GT.0) GO TO 700
      GO TO 710
700   QS = QS - QSINC
      QSINC = QSINC/10.
710   IF(QSINC.LT.0.1)GO TO 720
      QS = QS + QSINC
      BK = BKI
      IF(QS.GT.0.) CS = CSI
      IF(QS.GT.QSMAX) GO TO 100
        GO TO 1
720   QS = QS + QSINC*10.
      OMI = QS/100.
      IF(IPLOT.EQ.1) WRITE(2,999)FS,OMI
      ICOUNT = ICOUNT + 1
      WRITE(1,*)'OM:',OM,'  FS:',FS,'  FSMIN:',FSMIN
      WRITE(1,*)'  R1:',R1,RI1,'    R2:',R2,RI2
      WRITE(1,*)'   '
      WRITE(1,*)'             R0:',R0
      WRITE(1,*)'   '
      WRITE(1,*)'ROOT1:',RT1R,RT1I
      WRITE(1,*)'ROOT2:',RT2R,RT2I
      WRITE(1,*)'ROOT3:',RT3R,RT3I
      WRITE(1,*)'ROOT4:',RT4R,RT4I
      WRITE(1,*)'   '
      WRITE(1,*)'INSTABILITY FREQUENCY:',OMI
      WRITE(1,*)'-------------------------------------------------'
      BK = BKI
100      IF(FS.GE.FSMAX) GO TO 400
      IF(QS.LT.QSMAX) GO TO 199
      WRITE(1,*)'NO INSTABILITY FOUND'
      WRITE(2,999)FS,OM
      ICOUNT = ICOUNT + 1
199      FS = FS + FSINC
         QS = QSMIN
```

```
            QSINC = QSINCI
           GO TO 4
400         FS = FSI
            QS = QSMIN
            QSINC = QSINCI
1000     CONTINUE
         IF(IPLOT.NE.1)GO TO 9000
         ICOL = 2
         WRITE(3,998)ICOL,ICOUNT
         WRITE(3,997)'FSIDE'
         WRITE(3,997)'OMI'
999      FORMAT(2E11.4)
998      FORMAT(2I5)
997      FORMAT(A8)
         CLOSE(UNIT=2)
         CLOSE(UNIT=3)
9000     TYPE*,'FINISHED'
         CALL HORIZ(10,1)
         CLOSE(UNIT=1)
         STOP
         END

       SUBROUTINE HORIZ(I,N)
       IF ( I .EQ. 10 ) GOTO 100
       IF ( I .EQ. 16 ) GOTO 130
 10    FORMAT(10A1)
 100   WRITE(N,10) 27,91,49,119
       GOTO 200
 130   WRITE(N,10) 27,91,52,119
       GOTO 200
 200   CONTINUE
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       FILE NAME:  PNROOT.FOR          13-DEC-83       APW
C
C               NONDIMENSIONALIZED VERSION OF PSROOT.FOR
C
C       FUNCTION:  COMPUTES RO FOR VARIOUS SIDE FORCES, FREQUENCIES,
C                  AND DEADBANDS.  THE VALUE OF RO IS THEN USED TO
C                  DETERMINE THE ROOTS OF THE CHARACTERISTIC EQUATION
C                  WHICH HAS BEEN CAST IN THE FORM OF A PRODUCT OF
C                  TWO SECOND ORDER EQUATIONS IN 'S'.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        PROGRAM PSROOT
        DIMENSION G(6),GI(6),JDB(6)
        DATA GI /0.5E-3,1.0E-3,1.5E-3,2.0E-3,2.5E-3,0.0/
        BKI = 1.0E6
        SKI = 2.0E5
        CSI = 200.
        CQI = 40.
        RMI = 0.20422
        TYPE*,'INPUT THE NUMBER OF DEADBAND VALUES TO BE CONSIDERED,'
        TYPE*,'ONE TO SIX:'
        ACCEPT*,NDB
        IF(NDB.LT.6) GO TO 2
        DO 300 L = 1,6
300     G(L) = GI(L)
        GO TO 3
2       TYPE*,'INPUT THE SPECIFIC DEADBANDS DESIRED:'
        TYPE*,'(1).5E-3,(2)1.E-3,(3)1.5E-3,(4)2.E-3,(5)2.5E-3,(6)0.0'
        ACCEPT*,(JDB(K),K=1,NDB)
        DO 701 I = 1,NDB
        JI = JDB(I)
701     G(I) = GI(JI)
3       TYPE*,'INPUT FSI,FSMAX, & FSINC:'
        ACCEPT*,FSI,FSMA,FSIN
        TYPE *,'INPUT QSMIN,QSMAX, & QSINCI:'
        ACCEPT*,QSMIN,QSMAX,QSINCI
        TYPE*,'INPUT A "1" FOR OUTPUT TO BE PLOTTED.'
        ACCEPT*,IPLOT
CCC
C  NONDIMENSIONALIZE SYSTEM PARAMETERS
CCC
        TC = SQRT(RMI/(SKI+BKI))
        BK = BKI/SKI
        SK = SKI/SKI
        CS = CSI/(SKI*TC)
        CQ = CQI/SKI
        RM = RMI/(SKI*(TC**2))
        BKIG = BK
        CSIG = CS
        QSMIN = QSMIN/SKI
        QSMAX = QSMAX/SKI
        QSINCI = QSINCI/SKI
```

```
        QSINC = QSINCI
        BB = .10/SKI
CC      OPEN(UNIT=1,TYPE='NEW')
        IF(IPLOT.NE.1) GO TO 5
        OPEN(UNIT=2,TYPE='NEW')
        OPEN(UNIT=3,TYPE='NEW')
5       CALL HORIZ(16,1)
        DO 1000 K = 1,NDB
        FS = FSI/(SKI*G(K))
        FSMAX = FSMA/(SKI*G(K))
        FSINC = FSIN/(SKI*G(K))
CC4     WRITE(1,*)'G:',G(K),'    FS:',FS
CC      WRITE(1,*)'    '
4       CONTINUE
1       FSMIN = SQRT(SK**2 + QS**2)
        IF(FSMIN.GT.FS) BK=0.
        OM = QS*SKI*TC/100.
        IF(QS.EQ.0.) CS = 0.
           A = (BK + SK)**2 + QS**2
           B = (BK + SK)*BK
           C = FS**2*A - QS**2*BK**2
           R1 = B/A + SQRT(C)/A
           R2 = B/A - SQRT(C)/A
CCC
C   COMPUTE THE ROOTS OF THE CHARACTERISTIC EQUATION
CCC
C
C  DETERMINE RO
CCC
        IF(BK.EQ.0) GO TO 520
        IF(R1.GE.1.0)RO = R1
        IF(R2.GE.1.0)RO = R2
        GO TO 540
520     IF(R1.GT.0.) RO = R1
        IF(R2.GT.0.) RO = R2
CCC
C   COMPUTE DELTA -- IF DELTA IS COMPLEX, RK IS COMPLEX
CCC
540     DELTA = BK/(2.0*RM*RO)
        RK = (SK+BK)/RM - DELTA
CCC
C  COMPUTE THE DELTA-Q RADICAL
CCC
        RAD = DELTA**2 - (QS**2/RM**2)
        IF(RAD.LT.0.)GO TO 530
        DQREAL = SQRT(RAD)
        DQIMAG = 0.
        GO TO 550
530     DQIMAG = SQRT(ABS(RAD))
        DQREAL = 0.
550     XKRP = (RK + DQREAL)*4.0
        XKIP =  DQIMAG*4.0
        XKRN = (RK - DQREAL)*4.0
        XKIN = -XKIP
```

```
        REALP = (CS**2/RM**2) - XKRP
        REALN = (CS**2/RM**2) - XKRN
        IF(XKIP.NE.0.) GO TO 560
        IF(REALP.GE.0) GO TO 561
        XR1 = 0.
        XI1 = SQRT(ABS(REALP))
        GO TO 562
561     XR1 = SQRT(REALP)
        XI1 = 0.
562     IF(REALN.GE.0) GO TO 563
        XR2 = 0.
        XI2 = SQRT(ABS(REALN))
        GO TO 565
563     XR2 = SQRT(REALN)
        XI2 = 0.
        GO TO 565
560     SQMP = SQRT(SQRT(REALP**2 + XKIP**2))/2.0
        SQMN = SQRT(SQRT(REALN**2 + XKIN**2))/2.0
        SQMPT = (ATAN2(XKIP,REALP))/2.0
        SQMNT = (ATAN2(XKIN,REALN))/2.0
        XR1 = SQMP*COS(SQMPT)
        XI1 = SQMP*SIN(SQMPT)
        XR2 = SQMN*COS(SQMNT)
        XI2 = SQMN*SIN(SQMNT)
565     RT1R = - CS/(2.0*RM) + XR1
        RT1I = XI1
        RT2R = - CS/(2.0*RM) - XR1
        RT2I = -XI1
        RT3R = - CS/(2.0*RM) + XR2
        RT3I = XI2
        RT4R = - CS/(2.0*RM) - XR2
        RT4I = -XI2
        IF(RT1R.GT.0) GO TO 700
        IF(RT2R.GT.0) GO TO 700
        IF(RT3R.GT.0) GO TO 700
        IF(RT4R.GT.0) GO TO 700
        GO TO 710
700     QS = QS - QSINC
        QSINC = QSINC/10.
710     IF(QSINC.LT.BB)GO TO 720
        QS = QS + QSINC
        BK = BKIG
        IF(QS.GT.0.) CS = CSIG
        IF(QS.GT.QSMAX) GO TO 100
          GO TO 1
720     QS = QS + QSINC*10.
        OMI = QS*SKI*TC/100.0
        IF(IPLOT.EQ.1) WRITE(2,999)FS,OMI
        ICOUNT = ICOUNT + 1
CC      WRITE(1,*)'OM:',OM,'  FS:',FS,'  FSMIN:',FSMIN
CC      WRITE(1,*)'  R1:',R1,RI1,'    R2:',R2,RI2
CC      WRITE(1,*)'   '
CC      WRITE(1,*)'             RO:',RO
CC      WRITE(1,*)'   '
```

```
CC      WRITE(1,*)'ROOT1:',RT1R,RT1I
CC      WRITE(1,*)'ROOT2:',RT2R,RT2I
CC      WRITE(1,*)'ROOT3:',RT3R,RT3I
CC      WRITE(1,*)'ROOT4:',RT4R,RT4I
CC      WRITE(1,*)'   '
CC      WRITE(1,*)'INSTABILITY FREQUENCY:',OMI
CC      WRITE(1,*)'-------------------------------------------------'
        BK = BKIG
100        IF(FS.GE.FSMAX) GO TO 400
        IF(QS.LT.QSMAX) GO TO 199
CC      WRITE(1,*)'NO INSTABILITY FOUND'
        WRITE(2,999)FS,OM
        ICOUNT = ICOUNT + 1
199        FS = FS + FSINC
           QS = QSMIN
           QSINC = QSINCI
          GO TO 4
400     QS = QSMIN
        QSINC = QSINCI
1000    CONTINUE
        IF(IPLOT.NE.1)GO TO 9000
        ICOL = 2
        WRITE(3,998)ICOL,ICOUNT
        WRITE(3,997)'FSIDE'
        WRITE(3,997)'OMI'
999     FORMAT(2E11.4)
998     FORMAT(2I5)
997     FORMAT(A8)
        CLOSE(UNIT=2)
        CLOSE(UNIT=3)
9000    TYPE*,'FINISHED'
        STOP
        END

      SUBROUTINE HORIZ(I,N)
      IF ( I .EQ. 10 ) GOTO 100
      IF ( I .EQ. 16 ) GOTO 130
 10   FORMAT(10A1)
 100  WRITE(N,10) 27,91,49,119
      GOTO 200
 130  WRITE(N,10) 27,91,52,119
      GOTO 200
 200  CONTINUE
      RETURN
      END
```

6.0 Conclusions:

In the previous sections we have discussed in detail the modeling and analysis efforts under this study. It is now time to review and summarize our results.

1. Observed 3 motion types called A, B, C;

   A - Periodic but dues not enclose origin, may include higher harmonics;
   B - Nonperiodic;
   C - Periodic enclosing origin, synchronous or nonsynchronous;

2. Limit Cycle Algorithm developed and employed, both A & C types observed.

3. Deadband does not affect stability-in-the-large.

4. Stability-in-the-small are affected (enhanced) by deadband and sideforce.

5. Bearing loads are largest for C-type motion.

6. Side force acting in concert with deadband effects may either increase or decrease bearing loads.

7. Bearing loads in a stable pump are determined primarily by rotor imbalance and side forces.

These results are quite significant in our understanding of the effects of bearing deadbands. Harmonics of snychronous and nonsynchronous oscillations have been observed. This is clearly a nonlinear effect. Stable limit cycle whirls have been observed occurring at synchronous and nonsynchronous rotor speeds in our results.

The limit cycle algorithm that we have developed can be generalized to more complex turbopump models with more degrees of freedom. It will be useful for loads analysis with nonlinear forces for rotor dynamics and other applications. It is capable of converging to periodic motions (solutions) which generally result in the highest load-producing conditions.

Since stability-in-the-large is ultimately determined by behavior at extremely large amplitudes of motion, deadband effects become negligible. Thus, linear models remain adequate for analysis of global stability properties. Stability-in-the-small is significantly altered by the nonlinear effects of deadbands. We have shown that sideforces can significantly enhance stability provided imbalance offsets and/or impulsive disturbances do not cause significant displacement from the equilibrium position of the rotor.

Bearing loads have been shown to be significantly modified by deadband effects. Critical speeds are altered. Loads may increase or decrease. The shape of the critical response curve is altered with higher loading at lower frequencies due to the deadband.

These results have been obtained using a relatively simple 2 degree-of-greedom model. This may lead one to believe the results are not applicable to real machines. This is not the case, however, and indeed one can argue and demonstrate with more sophisticated models that these effects are real. Since rotor responses are most often periodic, such motions can be described adequately by an effective mass responding to effective stiffnesses and deadbands, i.e., a 2-dimensional model. Thus, our results are at least qualitatively valid for the description of turbo-pump motions.

Future work to be done in this area includes the generalization of our limit cycle algorithm to higher dimensional models including more nonlinearities. The thrust of this work would be to verify the results demonstrated for the Jeffcott model and explore the ways in which the additional degrees of freedom shift critical speeds around.

Additional work remaining to be done in this area involves the development of a simple test rotor whose dynamics would be adequately described by the Jeffcott model. Such a test rotor would allow accurate investigation of a number of rotordynamic phenomena. It would serve as an experimental tool for the refinement of our models of bearing forces including deadband effects. Once these forces become well understood, the test rotor could be used to explore and define more precisely such whirl driving mechanisms as fluid seals, impeller/diffuser coupling, rubbing between rotor and housing and rotor internal friction. The experimental research rotor would greatly improve theoretical understanding of rotordynamic forces and consequently whirl phenomena.

This has been extremely interesting and enjoyable work. We have appreciated the opportunity to contribute to the analytical state-of-the-art in this area.

REFERENCES

1. Childs, Dara W., Definition of Forces on Turbomachinery Rotors, Task A Report, prepared for G.C. Marshall Space Flight Center, Alabama 35812, under contract HAS 8-34505, p3.

2. Gunter, Edgar J. Jr., Dynamic Stability of Rotor-Bearing Systems, NASA SP-113; 1966, p 9ff.

3. Several G.C. Marshall Space Flight Center, ED14, Internal Memoranda and presenations covering period 1978-1980.